CRANFIELD UNIVERSITY

ARISTEIDIS TSITIRIDIS

BIOLOGICALLY – INSPIRED MACHINE VISION

Department of Informatics and Systems Engineering

Doctor of Philosophy
Academic Year: 2008- 2012

Supervisor: Dr. Mark Richardson
June 2012

“...For it is impossible for anyone to begin to learn that which he thinks he already knows.” – Epictetus, Greek philosopher, 55 - 135 AD

ABSTRACT

This thesis summarises research on the improved design, integration and expansion of past cortex-like computer vision models, following biologically-inspired methodologies. By adopting early theories and algorithms as a building block, particular interest has been shown for algorithmic parameterisation, feature extraction, invariance properties and classification.

Overall, the major original contributions of this thesis have been:

- The incorporation of a salient feature-based method for semantic feature extraction and refinement in object recognition.
- The design and integration of colour features coupled with the existing morphological-based features for efficient and improved biologically-inspired object recognition.
- The introduction of the illumination invariance property with colour constancy methods under a biologically-inspired framework.
- The development and investigation of rotation invariance methods to improve robustness and compensate for the lack of such a mechanism in the original models.
- Adaptive Gabor filter design that captures texture information, enhancing the morphological description of objects in a visual scene and improving the overall classification performance.
- Instigation of pioneering research on Spiking Neural Network classification for biologically-inspired vision.

Most of the above contributions have also been presented in two journal publications and five conference papers. The system has been fully developed and tested in computers using MATLAB under a variety of image datasets either created for the purposes of this work or obtained from the public domain.

Keywords: Computer vision, Computational neuroscience, Human vision models, Object recognition, Object classification

ACKNOWLEDGEMENTS

This thesis would never have been possible without the contribution from a number of individuals. First of all, I am grateful to my supervisor Dr. Mark Richardson for his constant encouragement and patient help through this thesis and an admittedly very stressful period of my degree and life. Furthermore, I would like to thank everyone at EPSRC who provided the necessary funds for this project. I would also like to thank Dr. Peter WT Yuen for materialising this project and his input on some sections of this project. In addition, I wish to express my appreciation to the professionalism, guidance and understanding I received from Cranfield University staff during my programme.

I would like to express further gratitude to a number of individuals who helped me during my degree in various ways. I thank Jim Mutch of the Massachusetts Institute of Technology, USA, for providing valuable advice on his model on a number of occasions; Fillip Ponulak of Princeton University, USA, for patiently answering my questions and for provision of his MATLAB code on RESUME and spiking neurons; Emmanuel Hourdakis of the University of Crete, Greece, for his help with liquid state machines and for provision of MATLAB code; Cornelius Glackin of Ulster University, UK, for his advice and provision of MATLAB code on fuzzy spiking neurons; Qingxiang Wu of Ulster University, UK, for provision of MATLAB code on spiking neurons; Lazaros Iliadis of Democritus University of Thrace, Greece, for his collaboration, patience and help.

Last but certainly not least, I wish to express my unconditional love and gratitude to my family and my partner Georgina, for all their help, understanding, advice and continuous support on all of my endeavours.

TABLE OF CONTENTS

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

1 INTRODUCTION

Mankind has always dreamed of crafting lifelike machines. Fictional ideas usually precede their scientific proof of concept and perhaps the earliest reference to such a fictional manlike self-operating machine (automaton) can be traced back to Talos, the mythical bronze statue constructed by the inventor and craftsman Daedalus, employed by Zeus to safeguard Europa from pirates in the island of Crete in Ancient Greece. Later accounts of creating automated machines can be found around the world with mixed elements of fiction and early science. The earliest evidence of such an attempt can be found from renowned inventor, artist and engineer Leonardo Da Vinci (1452 – 1519) who had designed a series of robotic devices including a humanoid machine [1]. It is not however until centuries later in 1954, that science provably caught up with these fictional concepts and the first working robotic design is created by George Devol for industrial use [2].

In later years, the growing needs in all aspects of modern life make biologically-inspired approaches increasingly attractive and practical. This is mainly because biology has solved or has adapted temporarily to (subject to continuous evolution) many real world problems in very efficient ways. Researchers across many scientific fields have thought of, or even employed, biological behaviour in their theories and applications. Biological aspects become even more attractive especially as the capabilities of computer hardware increase and are able to cope with the computational stress often associated with their implementation. For many scientists particularly in the fields of neuroscience, computer engineering and computer science, unlocking these biological mechanisms is not solely for understanding biology and in essence our very existence and self-awareness but also to tamper and improve the efficiency of both biological and artificial constructs.

The brain is the most complex organ of human biology and sits at the centre of the human nervous system with regions that are still not fully understood. It is a heavily studied part of the human body being responsible for a wide range of functions that stretch from cognition and reason to essential bodily functions such as movement. Naturally, all this processing comes at a certain price since almost 20% of the human body's energy is consumed by the brain. An adult brain has about one trillion cells out of which fifty to a hundred billion (5-10%) are estimated to be neurons alone. Broadly, an important characteristic of the human cortex is its ability to process information in a parallel and decentralised fashion, making it an efficient hub that can process, isolate and/or combine different sensory information such as sound, vision, touch etc. Another essential characteristic is its plasticity, i.e. its ability to "reprogram" itself in order to learn

and adapt to different requirements and conditions. Finally, its hierarchical structure (and to a certain extent modularity) ensures that low-level sensory information does not strain the whole brain with its existence and thus some cortical areas can focus unhindered on higher-level functions such as decision-making, thinking and multitasking.

Over millions of years human vision has been slowly evolving as part of the brain. Humans greatly rely on their vision for their survival and everyday activities. From driving a car to watching television, from deciding what to wear everyday to finding food and surviving in the wild, vision has always been a very rich and important source of environmental information. Perhaps it is of no surprise that all the wealth of light information has forced the cognitive visual system and the compartments of the brain responsible for visual processing to utilise nearly 40% of the brain's volume. All these billions of neurons work collectively to produce their remarkable results, for example in the primary visual cortex alone it is estimated that about 140 million neurons receive and process in unison the raw visual stimuli from the optic tracts as transmitted from about 125 million photoreceptors from each of the eyes' retinas. Humans are on average effortlessly capable of detecting, recognising or even mentally visualising, synthesising and projecting objects in just milliseconds. Furthermore, they are capable of doing that irrespective of an object's size, position, orientation, illumination and depth projections. These generic visual properties often termed as object constancy, occur as the brain shields the upper cortical layers of consciousness from their existence and thus humans are unaware of their presence. The abilities of human visual cognition may excel and in fact be the best-known for object constancy, memory of thousands of objects, visual events, places and faces, complex associations and patterns between objects and actions, but human eyesight performance is task-dependant and is overall of course suitable for human activities. Generally, animal eyesight has adapted according to the nature and requirements of each species. For example, specific birds of prey such as eagles, hawks etc surpass primate eyesight in visual acuity and spectral perception. Other birds of prey like owls, being mostly hunters at night, have developed extreme night-vision abilities, whereas many insect species have compound eyes for activities that require acute peripheral vision.

Mimicking the brain to suit a particular task is the "holy grail" for biologists, neuroscientists, engineers and computer scientists alike. Such research opens up a fascinating world that diverges or even complements manmade techniques depending on task and application. Neuromorphic hardware architectures such as the Modular Neural Exploring Traveling Agent (MoNETA) [3] which is a memristor-based brain-inspired microprocessor and Neurogrid [4], another

analogue chip development that similarly simulates neuronal activity, both pave the way towards biological-like “intelligent” microprocessors that in the future might substitute traditional microprocessors. An attempt on the very topic of consciousness is made with the Blue Brain project [5], an ambitious neo-cortex simulation that if successful in the future, might even take on the monumental task of simulating an entire brain.

Overall, biological vision is a vast and challenging research topic which remains to this day largely unexplored. Its principles have many overlapping fields such as neuroscience, cognitive psychology, artificial intelligence, physics, hardware and software engineering. This project resides primarily within computational neuroscience and image processing with elements from all aforementioned scientific fields.

1.1 Aim

This project has been successful in its primary objectives which were to:

- Explore and outline previous relevant research work across all the associated fields.
- Develop a biologically-plausible visual software model, following the dominant models and research in the area.
- Enhance the newly developed model's behaviour by implementing as many of the biologically-inspired notions of object constancy as possible.
- Gradually develop methods to enhance biological realism, design and overall performance by examining physiological evidence.
- Identify critical issues affecting the model's performance and investigate methods to expand future research.
- Lay the groundwork for a full future transformation of the developed biologically-inspired vision techniques to neural networks.

Naturally, at this stage the methodologies developed in this project do not claim a full biological architecture and construction. In the absence of sufficient physiological/neuro-scientific evidence and given the project's time constraints and resources, hypotheses are applied and mathematically conceived techniques are employed where necessary. Nevertheless, the key objective of establishing a road map towards biological plausibility is pursued throughout this thesis.

1.2 Contributions and achievements

The research described in this thesis produced the following original research contributions.

- The incorporation of a salient feature-based method for semantic feature extraction and refinement in object recognition. This particular contribution is analysed in detail in section 6.1.1 of this thesis.
- Having increased the performance of the model via saliency, further enhancements on the existing architecture are gradually added in sections 5.4 and 6.1.2.
- The design and integration of colour features coupled with the existing morphological-based features for efficient and improved biologically-inspired object recognition. This new colour and morphological based design is constructed together with visual attention and examined under two different techniques. The theoretical and experimental analysis of these techniques is presented in section 6.2.
- An illumination invariance property using colour constancy under a biologically-inspired framework is presented and explored in section 7.1.1.
- The performance of the colour constancy method is measured with popular colour constancy techniques both individually (section 7.1.1) and in a fused approach (section 7.1.2).
- The development of rotation invariance methods in an attempt to improve robustness and compensate for the lack of such an important mechanism in the original models. This contribution is investigated throughout section 5.4.
- Adaptive Gabor filter designs that capture texture information, enhancing the morphological description of objects in a visual scene and improving the overall classification performance against a wider variety of image databases. The novel comparative study of the different Gabor filter designs is detailed in section 7.2

This project has produced in total 5 conference publications, 2 journal publications and has given the author the opportunity to become an academic publication reviewer for the Elsevier journal “Neurocomputing”, “Artificial Intelligence Applications and Innovations” – AIAI 2012, and for IEEE events such as IEEE Conference on Control, Systems & Industrial Informatics, IEEE Symposium on Business, Engineering & Industrial Applications, IEEE Symposium on Industrial Electronics and Applications.

2 THE NEURON

The fundamental knowledge behind the biology and operation of neurons and neural networks is established in this chapter. The neuron, being the building block of all information processing occurring in the brain and the nervous system, deserves special attention particularly as this thesis is concerned with its operation and implementation. Section 2.1 starts with the biology of neurons and section 2.2 explains Spiking Neural Networks as a direct application of biological neurons and covers issues often found in their implementation.

2.1 Neurons, synapses and neuronal circuits

A nerve cell known also as a neuron or neurone, has the role of transmitting information throughout the body by receiving electrical and chemical stimuli from adjacent input cells. Processing is achieved in various ways e.g. summation, maximum response or integration, and the neuron transmitting information is called a presynaptic cell and others that may follow, postsynaptic cells. Neurons have very sophisticated structures (Figure 2-1 and Figure 2-2) that can be broadly separated into the following sections which are analysed further in this chapter:

- Synapses
- Dendrites
- Cell body (Soma or perikaryon or cyton)
- Axon and axon terminals

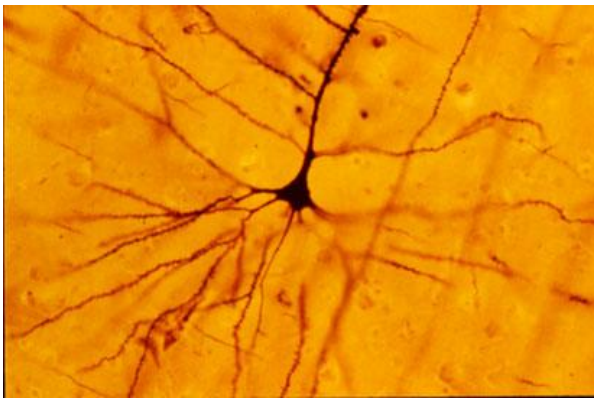


Figure 2-1: An example of a Golgi stained pyramidal neuron [6]**.**

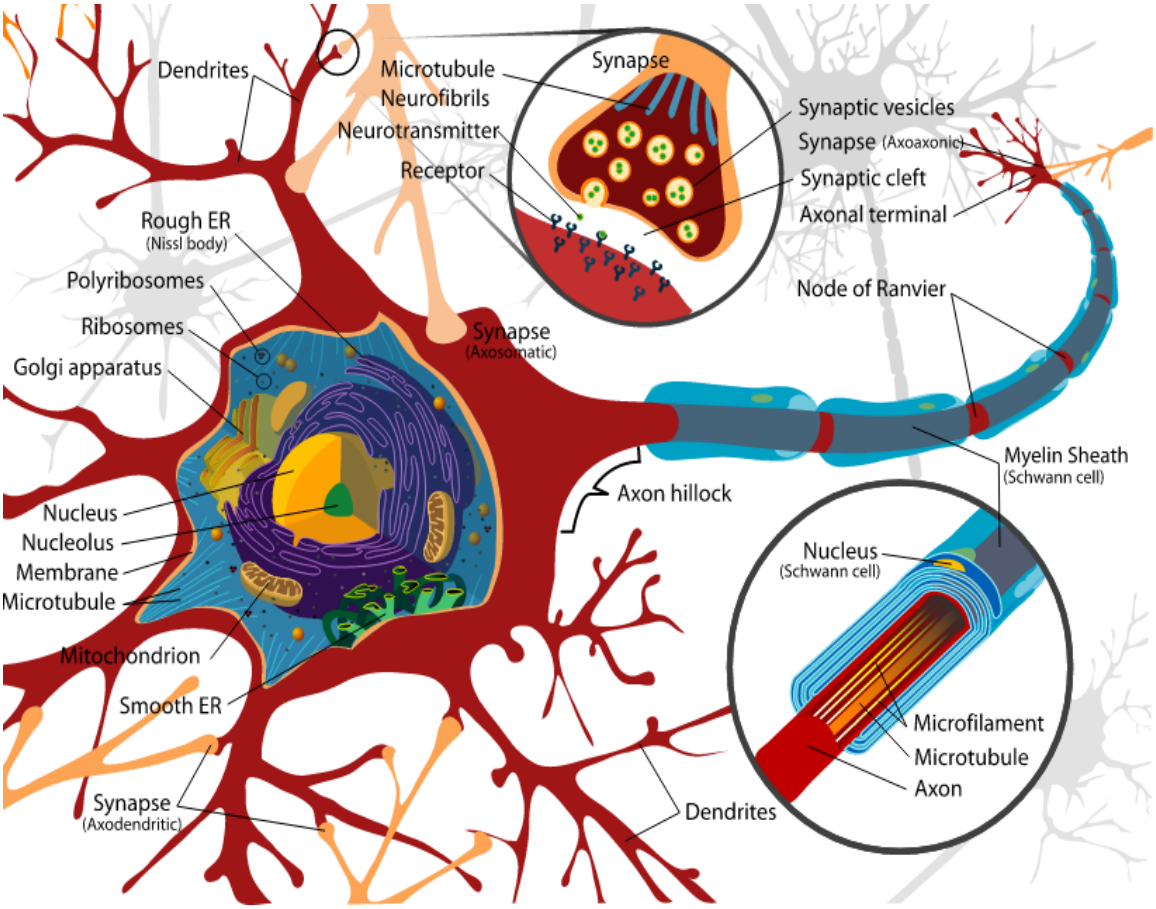


Figure 2-2: An illustration of a typical neuron's structure [7]**.**

2.1.1 Synapses

Synapses are the junctions at which an electrical or chemical impulse is conveyed between neurons (as shown in Figure 2-2). Their surface area is approximately 0.5 -2μm with the presynaptic terminal always being somewhat larger [8]. Synapses can be separated into two types, excitatory (also known as type 1, with small round vesicles as in Figure 2-2) and inhibitory (also known as type 2 with small polymorphic vesicles). Generally, an excitatory synapse increases the chance of an impulse (electrical stimulation) being propagated from the presynaptic to the postsynaptic neuron. On the other hand, an inhibitory synapse decreases this chance. There are more excitatory synapses (80-85%) than inhibitory (15-20%) and on average there are a total of 6000-7000 synapses per neuron giving rise to approximately 100-500 trillion for an adult human brain.

Excitatory and inhibitory synapses have each two methods of signal transmission, electrical and chemical. The electrical synapse, with virtually no delay, directly transmits electrical stimuli by electrical conductance over a very narrow synaptic cleft (or gap junction) of approximately 3.5 nm without any

amplification. Electrical synaptic clefts have minimal resistance over the electrical currents that flow from presynaptic to postsynaptic nerve cells [9]. Information over electrical synapses is usually bidirectional and such types of synapses are mostly found in areas associated with movement and reflexes where quick responses are essential.

A chemical synapse has usually a larger cleft of 20-40 nm with a delay that varies between 0.3-5ms and in contrast to the electrical, it amplifies signals [9]. Rather than electricity directly, a chemical synapse uses neurotransmitters i.e. protein elements such as amino acids, via a process called exocytosis. After receiving an action potential or impulse and according to its frequency or timing, an excitatory chemical synapse increases secretion of neurotransmitters contained in its vesicles (sacks). In turn, the rate of neurotransmitter fusion on the cell's membrane activates an influx of calcium (Ca^{2+}) ions through voltage-gated ion channels i.e. channels that open or close according to the potential difference between extracellular and intracellular space. As neurotransmitters are emitted from the presynaptic cell, they attach to receptor channels on the postsynaptic side causing sodium channels there to open. By allowing sodium (Na^{+}) ions to flow in, an action potential is created on the postsynaptic side (Figure 2-3). Vice versa an inhibitory synapse utilises inhibitory transmitters to decrease the chance of this phenomenon occurring or preventing it altogether [9]. Consequently, at the postsynaptic cell the nature of the synapse gives raise to either an excitatory postsynaptic action potential (EPSP) or an inhibitory postsynaptic action potential (IPSP). These action potentials are the electrical impulses of information between neurons that are described in more detail below.

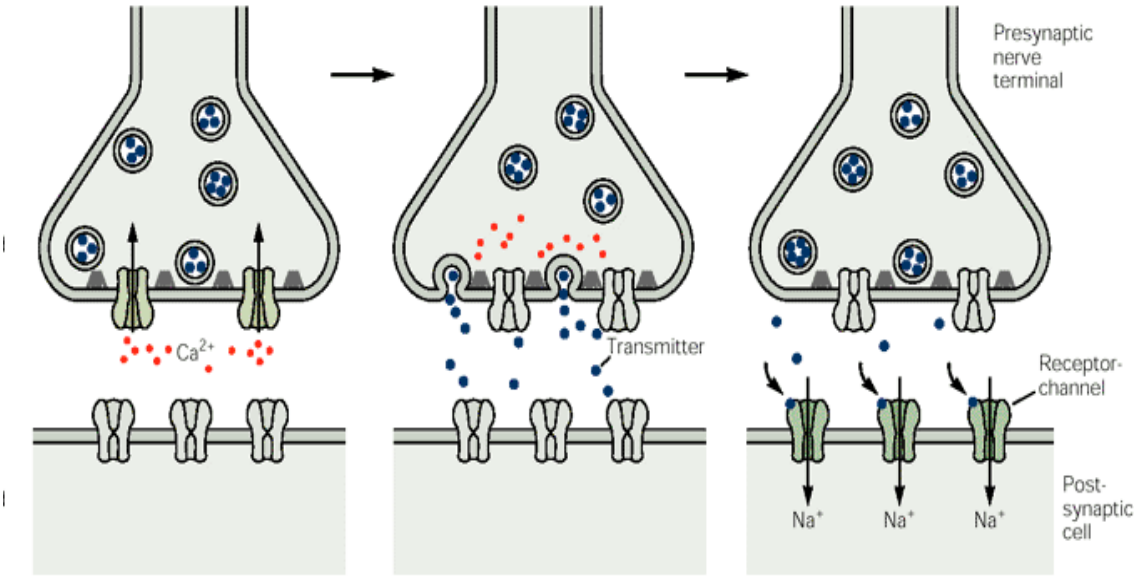


Figure 2-3: The process of a chemical synaptic transmission. From left to right, the action potential causes the calcium ion channels (Ca^{2+}) to open as vesicles fuse on the cell membrane releasing neurotransmitters that diffuse across the synaptic cleft eventually binding with receptors at the postsynaptic cell's sodium channels (Na^+). This process opens or closes the sodium channels and releases or prohibits the release of an action potential at the postsynaptic cell. It is this process that causes the delay in transmission [9]**.**

As importantly, there are various patterns of connectivity (microcircuits) in synapses that relay information via several ways for different reasons. These are broadly categorised as [8]:

- Synaptic divergence when a signal propagates through two or more synapses.
- Synaptic convergence when two or more neural signals may have an influence on one neuron input.
- Presynaptic inhibition when a type of convergence where a synapse may inhibit another before the postsynaptic nerve cell.
- Feedforward inhibition when inhibition flows in one direction.
- Recurrent or feedback inhibition when inhibition after an EPSP inhibits the presynaptic cell from firing again.
- Lateral inhibition.

Lateral inhibition plays a crucial role in many functions of the brain and for human vision in particular, it ensures that the antagonistic competition between photoreceptors in the retina enhances edges and contrast. It can in other words be expressed as a noise suppressing mechanism that at the same time promotes peak responses [8–10]. Lateral inhibition is examined further in section 3.1. Finally, microtubules and neurofibrils (or neurofilaments) shown in Figure 2-2, are scaffolding type fibres that not only provide support and structure for the entire nerve cell but also transport important proteins for cell health and communication.

2.1.2 Dendrites

Dendrites are branched extensions (often termed also as postsynaptic connections) of the neuron's soma that project outwards receiving information from other connected neurons. Many of their known properties were explored in a pioneering study by Wilfrid Rall [11]. Most dendrites carry tiny spines (approximately 0.5 – 6μm in length, 0.1 – 2μm in width) that based on their morphology can be very loosely fitted into five categories: thin, mushroom, filopodium, cup-shaped and stubby [12]. The varying shapes of spines control the synapse (connectivity) strength between different neurons and this is of critical importance in the transmission of stimuli between neurons. Furthermore, it is these spines that incorporate the plasticity term mentioned in chapter 1 since their formation, elimination, persistence and shape is directly linked to the novel or long-term experiences of animals, memory removal or even learning and relearning tasks [13]. In other words, they establish the necessary neuronal pathways for what is referred generally as memory (summarised in Table 2-1).

Synaptic Activity	Duration	Induction position
Short term Enhancement		
Paired-pulse facilitation (PPF)	100ms	Pre
Augmentation	10s	Pre
Post-tetanic potentiation	1min	Pre
Long-term Enhancement		
Short-term potentiation	15min	Post
Long-term potentiation	>30min	Pre and Post
Depression		
Paired-pulse depression (PPD)	100ms	Pre
Depletion	10s	Pre
Long-term depression (LTD)	>30min	Pre and post

Table 2-1. Different kinds of synaptic plasticity. Facilitation and potentiation refer to a synapse increasing its probability of transmitting an action potential whereas depression refers to the opposite.Terms pre and post refer to presynaptic and postsynaptic connections respectively [14]**.**

2.1.3 Cell body

A neuron's main body (soma), being the processing plant of many operations, is by far the most complex part of the neuron. It constitutes, as shown in Figure 2-2, a membrane bound nucleus and a series of organelles. The soma is both a protein synthesis plant and a generator of energy. Within the soma Nissl granules, ribosomes and ribosome clusters (polyribosomes or polysomes) decode Ribonucleic Acid (RNA) instructions from Deoxyribonucleic Acid (DNA) in order to create vital protein compounds. Via the Golgi apparatus these proteins are dispatched around the cell for processes such as exocytosis [9]. Another important organelle is the Mitochondria which are responsible for processes such as cell death, cell development, but more importantly for producing energy [15].

A cell body's membrane properties were first revealed by Alan Hodgkin and Andrew Huxley [16] which further led to the creation of their famous model, which is analysed in section 2.2.4. The membrane (or plasma) surrounds the neuron acting as a barrier between the core and the space around it. It exhibits certain properties among others, such as diffusion (molecules, ions diffuse and react with specialised receptors) and endocytosis (direct absorption of molecules) and more importantly for the context of this work, it possesses electrical properties for information transmission and intercellular communication. More specifically, during its equilibrium (idle) state a membrane has a resting potential, the voltage difference between the intracellular and immediate extracellular matter, of about -70mV to -75mV. In fact, this resting

potential is an average value only for certain pyramidal and primary visual cortex cells, as an example, retinal cells have a resting potential of -40mV [17].This equilibrium potential (E) can be calculated from Nernst's equation as:

$$E_X = \frac{RT}{zF} \ln \frac{[X]_o}{[X]_i} \quad \textbf{(2-1)}$$

In equation (2-1), R is the gas constant [8.315 J per Kelvin per mole (J K−1 mol−1)], T is temperature in Kelvin, F is Faraday's constant [96,485 coulombs per mole (C mol−1)], z is the valence of the ion, and $[X]_o$ and $[X]_i$ are the ion concentrations outside and inside the membrane.

The reduction of the resting potential is known as depolarisation and the opposite event of increasing it, hyperpolarisation (Figure 2-4). Depolarisation increases the likelihood of a neuron generating an impulse and is therefore linked with excitation whereas hyperpolarisation with inhibition. During depolarisation for example, inside the membrane there is a higher concentration of negatively charged molecules such as proteins and as soon as a synaptic event (Figure 2-3) causes sodium channels to open at the postsynaptic cell, then positively charged sodium ions (Na^+) flow in from the outside in order to balance out the potential difference (rising phase, Figure 2-4). At the same time, more potassium channels start to open (a certain number of potassium channels are always open) releasing potassium ions (K^+) out of the membrane causing repolarisation (failing phase, Figure 2-4), past the resting potential value (afterhyperpolarisation) ensuring the membrane's inactivity for 1-2 ms. This procedure produces an action potential. It is important to note here that if the threshold value is not surpassed by the stimulus of synapses (failed initiations, Figure 2-4) then the state of the neuron remains unchanged, known also as "all-or-one response". The failed initiations are simply products of the constant diffusion of ions that occurs in the membrane.

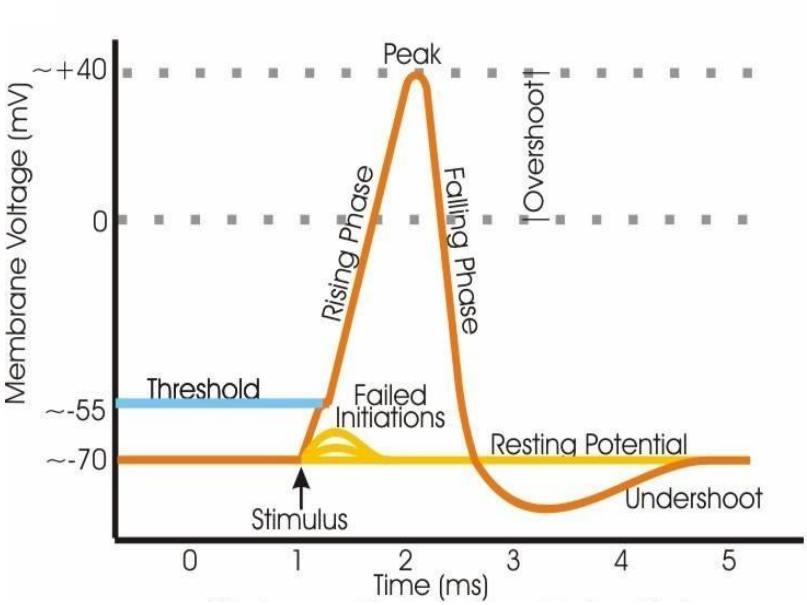


Figure 2-4: A schematic illustration of an action potential in which a resting potential of -70mV receives a stimulus that drives the process of depolarisation (rising phase) after the threshold has been exceeded. Repolarisation occurs when the maximum value has been reached (falling phase) and afterhyperpolarisation (or refractory period, i.e. a recovery period of forced inactivity) takes place after the potential is below the initial resting potential value (undershoot) [18]**.**

Conversely, when hyperpolarisation (inhibition) occurs, the stimulus of a synapse causes the resting potential of the membrane to increase inducing its inactivity for a given amount of time until equilibrium is again reached.

Neurons come in all shapes and sizes (4 – 100μm) [9], with numerous functions to perform in the central and peripheral nervous system. Hence, they can be classified according to their:

- Structure (e.g. unipolar, bipolar and multipolar).
- Function (e.g. afferent, efferent and interneurons).
- Discharge patterns (e.g. fast spiking, regular spiking, and bursting).
- Neurotransmitter production (e.g. cholinergic, glutamatergic, dopaminergic neurons).

2.1.4 Axons and axon terminals

Lastly, neurons transfer their action potentials or spikes through axons which vary in diameter from 1-20μm and in length from 0.1 mm to 3 metres [9]. Axons

are in intervals coated with myelin sheaths, a fatty substance which electrically insulates the transmission of spikes that travel down the axons as fast as 120 meters/second (Figure 2-2) [9]. Bundles of axons are joined together in larger nerve fibres. Nodes of Ranvier are amplifying gaps along the axon length, i.e. exposed areas allowing ion exchange, which propagate spikes down to the axon endings or axon terminals. The axon terminals eventually branch out to form dendrites and thus synaptic connections with other postsynaptic cells.

Neurons are organised in large clusters and consecutive layers designed to complete specialised tasks. Generally, these neural regions are classified between the central nervous system (also known as CNS and includes the brain, spinal cord, retina) and the peripheral nervous system (PNS) which connects neurons and nerves around the body with the CNS. Specifically in this work, efforts are concentrated on the biological vision and pattern recognition processes of the CNS.

2.2 Spiking Neural Networks

Spiking Neural Networks (SNN), also less popularly known as Pulsed Neural Networks, are models of Artificial Neural Networks (ANN) which are the most biologically plausible to-date. Their plausibility arises from their temporal nature and biological-like properties unlike other static models of ANN which in some cases show questionable relevance to biology e.g. Kohonen maps. SNN simulate a diverse number of neuron behaviours, individual neurons only activate when certain criteria have been met, i.e. the simulated membrane potential value has been exceeded to generate a simulated pulse or spike. Therefore, information in SNN is encoded and conveyed in the form of spikes and long sequences of spikes are termed spike trains. It is their biological realism along with their computational power and non-linearity that makes SNN particularly theoretically attractive and potentially practical for a variety of applications.

2.2.1 Artificial Neural Networks – An overview

The history of Artificial Neural Networks (ANN) begins with Alexander Bain as early as 1873 [19] who theorised that activities in succession activate sets of neurons and that repetition strengthened memory. Years later in 1943, McCullouch and Pitts [20] laid the scientific foundations by formulating a mathematical model for “threshold” neural networks. Later that decade in 1948, Alan Turing, who famously decoded the Enigma machine during the Second World War, intensified efforts on simulating binary neural networks and in 1949, Donald Hebb published a learning mechanism simulating the neurological learning process, also known as Hebbian learning (section 2.2.8) [21]. In the

following decades, concentrated efforts in this area produced significant concepts and some successful applications, such as the Perceptron, the Cognitron, the Hopfield network and the Backpropagation Algorithm which amplified the theoretical interest in ANN.

Today, there are countless ANN models or their variants used in a wide selection of applications in scientific and industrial areas. The following subsections, focus on temporal models

2.2.2 Integrate-and-fire

Arguably the earliest attempt to model a biological neuron was introduced in 1907 by Louis Lapique [22]. This model belongs to the category of thresholded neuron models and is known as integrate-and-fire (IF). More specifically, it is simply represented by the use of a capacitor. The capacitor's current-voltage time relationship is given by:

$$i(t) = C\frac{dv(t)}{dt} \qquad \textbf{(2-2)}$$

In equation (2-2), when a current is applied to a capacitor the voltage (membrane potential) increases with time reaching a certain threshold value at which it discharges and returns back to a resting potential. This is a generalised concept of how a neuron produces a spike and so it does not take into account several other features of the original biological procedure. Equation (2-2), expresses a linear relationship which cannot be applied to the biological. Furthermore, in section 2.1 it was seen that the neuron has an all-or-none principle whereby the values below the threshold are not retained. This notion is not expressed by equation (2-2) since individual successive current inputs accumulate to produce a spike.

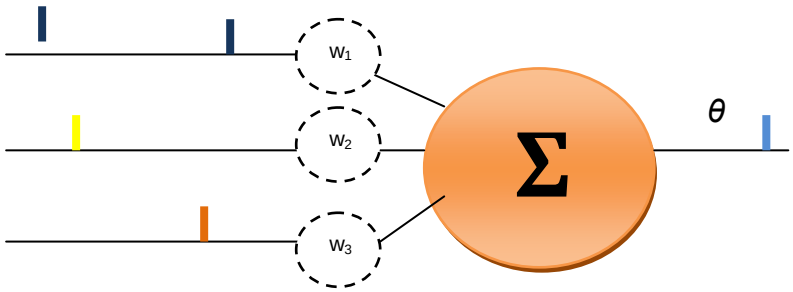


Figure 2-5: An illustration of an integrate-and-fire neuron. Spikes represented by vertical bars with associated weights (w_1, w_2 and w_3) accumulate to exceed a threshold value and thus emitting a resultant spike train.

2.2.3 Leaky integrate-and-fire

Following the absence of some biological notions in the integrate-and-fire model, modifications extended this model for biological plausibility without increasing its complexity. By adding a “leaky” part then successive currents should not exceed the threshold value and yield a spike but rather diffuse as ions would in an actual neuron. This leaky term is introduced with a resistance thus creating an RC electrical circuit:

$$i(t) - \frac{v(t)}{R} = C\frac{dv(t)}{dt} \qquad \textbf{(2-3)}$$

Or

$$i(t) = C\frac{dv(t)}{dt} + \frac{v(t)}{R}$$

Substituting in (2-3) for time constant τ_m = RC yields:

$$\tau_m \frac{dv(t)}{dt} = -v(t) + Ri(t) \qquad \textbf{(2-4)}$$

Equations for a Leaky integrate-and-fire (LIF) neuron, have introduced the biological concept of ion diffusion across the membrane. However, being first order linear differential equations means they do not address the non-linearity aspect of biological neurons, e.g. a typical LIF representation is depicted in Figure 2-6. For this reason, further non-linear versions have been developed such as the Non-linear Integrate-and-Fire (NLIF) or Quadratic IF (QIF), the Exponential NLIF and Multicurrent IF [23].

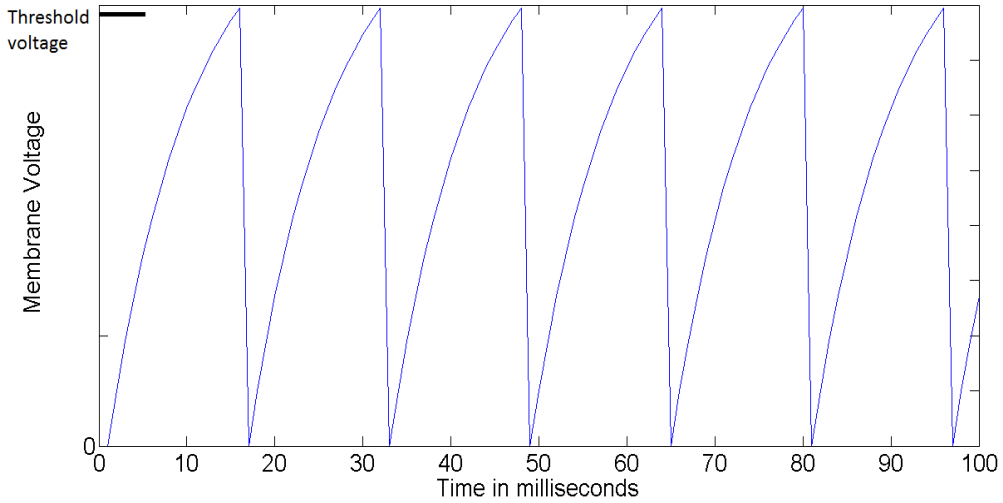


Figure 2-6: An example of a LIF neuron typical response with respect to time.

2.2.4 Hodgkin-Huxley

Alan Hodgkin and Andrew Huxley (HH) [16], proposed a realistic model after experiments conducted on a giant squid axon. Their model belongs to another category of spiking neural model based on conductance. The HH model describes the mechanisms of passive (leaky), sodium (Na) and potassium (K) ion channels. The membrane potential is characterised by the total of the currents applied on these ion channels removed from the overall current as it entered from extracellular space.

$$C\frac{dv(t)}{dt} = I(t) - \sum_k I_k \qquad \textbf{(2-5)}$$

In equation (2-5), V is the membrane voltage, I_k are the ionic channel currents and $I(t)$ is the total applied current. Furthermore, these ionic currents are shown to depend on a number of conductances g_{Na}, g_k and g_L for the sodium, potassium and leakage channel respectively.

$$\sum_k I_k = g_{Na} m^3 h(v - E_{Na}) + g_k n^4 (v - E_k) + g_L (v - E_L) \qquad \textbf{(2-6)}$$

Gating variables m, n and h from equation (2-6), control the probability that channels are open at a time t, equations (2-7), (2-8) and (2-9). If all channels are open, currents are transmitted with maximum conductances g_{Na}, and g_k. Reversal potentials E_{Na}, E_k and E_L and functions α, β, are empirical observations from their experiments:

$$m = \alpha_m(v)(1-m) - \beta_m(v)m \quad \textbf{(2-7)}$$

$$n = \alpha_n(v)(1-n) - \beta_n(v)n \quad \textbf{(2-8)}$$

$$h = \alpha_h(v)(1-h) - \beta_h(v)h \quad \textbf{(2-9)}$$

Equations (2-5) to (2-9), characterise the Hodgkin - Huxley model which has been the subject of much research and the foundation for other more recent conductance based models such as the FitzHugh-Nagumo [24], Morris-Lecar [25] and Hindmarsh-Rose [26]. An illustrative example of the Hodgkin – Huxley model is shown in

Figure 2-7 and the associated response in Figure 2-8.

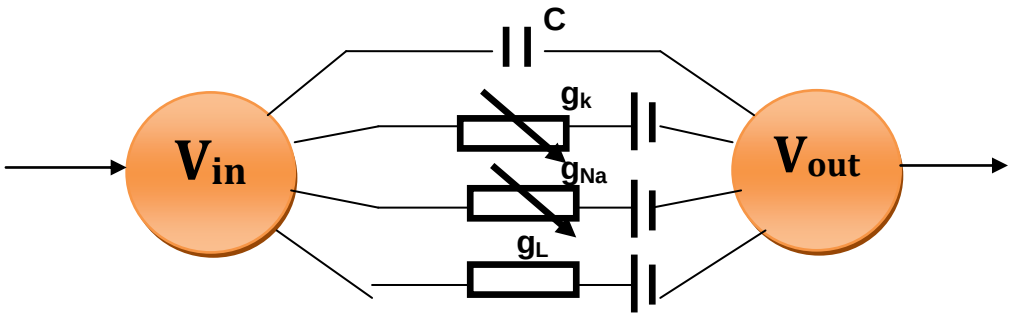


Figure 2-7: An illustrative example of the HH model. An extracellular potential enters passing through the leakage channel G, the gated potassium channel G_k and the gated sodium channel G_{Na}, as they alter the membrane potential.

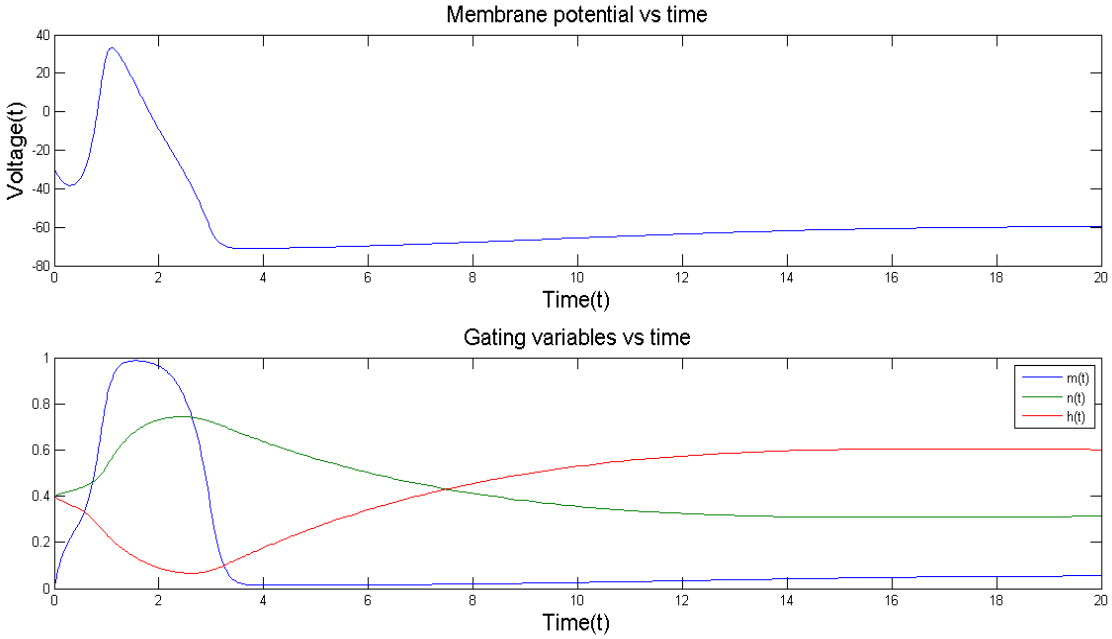


Figure 2-8: An example of a HH neuron typical response using MATLAB. Top graph shows the activity of the membrane potential with respect to time. Bottom graph illustrates the behaviour of the three gating variables with respect to time.

2.2.5 Synaptic transmission

In section 2.1.1 of this chapter, the biological synaptic process was described along with its significance to mechanisms of plasticity, memory, learning and spike transmission. In artificial neural networks, the synaptic properties are often oversimplified by being equated to static weights. This is far from ideal or realistic and by introducing realistic variables, the synaptic process becomes more computationally powerful and efficient [27].

In a dynamic synapse model, the state of the synapse changes over both time and input. The first spike that arrives at a particular synapse is assumed to be transmitted at full efficacy and any subsequent spike transmission would depend on the amount of neurotransmitters present [27]:

$$A_n = A \cdot u_n \cdot R_n \qquad \textbf{(2-10)}$$

In equation (2-10), A_n is the weight or efficacy associated to a synapse transmitting the nth spike and absolute synaptic efficacy A is affected by variables u_n and R_n that control the fraction of synaptic resources (neurotransmitters). Specifically, u_n is the facilitating fraction of synaptic efficacy while R_n is the depressing fraction of the synaptic efficacy and they are

inversely connected so that, at the first spike, $R_1 = 1- u$ [28]. Subsequent to the n^{th} spike, recovery of the synapse efficacy varies with time such that:

$$u_{n+1} = u + u_n(1-u)\exp(\frac{-\Delta t_n}{F}) \qquad \textbf{(2-11)}$$

$$R_{n+1} = 1 + (R_n - R_n \cdot u_n - 1)\exp(\frac{-\Delta t_n}{D}) \qquad \textbf{(2-12)}$$

F is the factor associated with synapse recovery from facilitation (weight increase) and D as the factor associated with synapse recovery from depression (weight decrease). In other words, equation (2-12) shows the fraction of synaptic efficacy available after the n^{th} spike at time Δt_n. Equations (2-11) and (2-12), describe the time relationship of synaptic efficacy in a dynamic synapse through facilitation and depression.

2.2.6 Spike coding

So far in this chapter, the biological processes behind the transmission of information have been covered by describing the operation of the biological neuron and some of its direct simulations. However, crucial questions arise about the techniques neurons use to encode-decode information and their possible implementation.

Research on this topic is still active and there is some controversy surrounding the exact nature of the biological principles but certain conclusions about neuronal coding methods have been made. One important characteristic is that biological neurons code action potentials or spikes based on time. These time coding methods are not exclusive and highly depend on the neuron's purpose or assigned tasks. In 1926, Adrian and Zotterman proved the existence of a firing rate (rate coding) mechanism by applying different stimuli on the sensory nerves of frogs [29] while some years later in 1929, Troland showed that in auditory cognition spike transmission may be expressed via its exact timing (temporal order coding) [30].

Rate coding methods include methods of examining the average number of spikes in a given time window, an average over repeated responses or a population coding relationship as found in groups of motor cortical cells [23] [31]. Under this population method in particular, each individual neuron fires at a high rate (vote) for a favoured direction and a lower rate in others. The final vote is obtained by vector contributions of all cells together to produce a "neuronal population vector" or population code.

Many recent neuroscience studies have demonstrated the existence of precision temporal order coding schemes in higher cognitive functions such as in the visual cortex of a blowfly [32], the rabbit retina [33] and the owl's auditory cortex [23]. Results of a comprehensive practical evaluation for retinal ganglion cells have shown that optimal performance is achieved using a temporal ranking method [34]. Methods of temporal ranking include [23]:

- Time-to-first spike: An approach that encodes information on the time required for the first spike to appear.
- Phase encoding: Produced spikes follow the phase oscillations of periodic analogue signals.
- Rank encoding: Spikes are encoded according to their order of appearance in a given time window.
- Latency encoding: Precision spike timing.
- Correlations and Synchrony: Neurons spike at correlated distinct patterns e.g. sparse coding [35] or synchronous spike emission for a given event.
- Reverse Correlation: Stimulus-driven spike generation.
- Linear encoding: Direct linear transformation of vectors or signals into a time scale [36].

None of these schemes can be considered ideal or universal and the choice largely depends on the nature of the task that spiking networks have to perform.

2.2.7 Liquid State Machines

Neural network architectures define the structure and arrangements that artificial neurons have from input to output. In some networks, inputs are directly connected to outputs while in other architectures more effectively, hidden layers between inputs and outputs, process information in an attempt to isolate patterns within it. However, adding hidden layers is not a straightforward and intuitive task since using fewer neurons for a given hidden layer, causes underrepresentation (underfitting) while an excessive number of neurons causes overrepresentation (overfitting) of the input data. Both, lead to poor network performance and generalisation properties (as represented in Figure 2-9). There have been remedies developed against underfitting and overfitting but it is actually the fixed structure of a network that contributes to the problem itself. In the biological brain, the number of neurons assigned for a task is in fact

never the same and it is such dynamic structures that can outperform fixed topology networks.

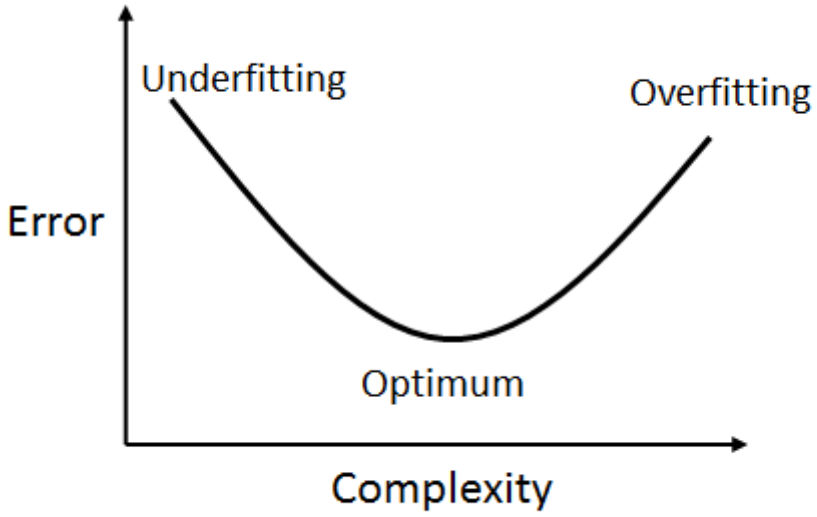


Figure 2-9: An illustration of the error versus complexity relationship in ANN

Spiking neural network architectures are divided like other types of ANN, in three categories based on the direction of flow of the data within them:

- Feedforward spiking neural networks (FSNN), in which neurons propagate in only one direction from input to output such as in Bohte et al [37].
- Recurrent spiking neural networks(RSNN) in which neurons can interact via feedback connections such as echo state networks or as in Tino and Mills [38].
- Hybrid spiking neural networks (HSNN) in which neurons may exhibit elements of feedforward and recurrent characteristics such as the Liquid State Machine (LSM) [39].

In an attempt to overcome fixed topology drawbacks, evolving spiking neural networks (ESNN) instead have an adaptive topology and several optimisation methods have been employed to accomplish this, such as Genetic Algorithms (GA)[40], Particle Swarm optimisation [41], in hardware via Field Programmable Gate Arrays (FPGA) [42] or utilising learning accuracy as provided by a distance metric, as shown in Wysoski et al [43], [44]. All these methods stem

from the notion of topology evolution but either lack universality or biological plausibility to be used as a standardised optimisation method or framework for SNN topology.

LSM (also known as Neural Microcircuits or NMC) are a hybrid framework for spiking neural networks and belong to a concept called reservoir computing that provides, according to Maas [39], universal computation power. In LSM, similarly to ripples on the surface of a liquid substance such as water, information about the origin and timing of these ripples (spiking events) can be extracted. This liquid concept can also be viewed as an echo, hence the similarity between LSM and echo state machines, of a past event or in other words as memory. If the origin of the rippling event is assumed to be some continuous time function *u(t)* then it can be applied as input to a liquid medium L^M, where *M* symbolises the different maps of a stream. The output stream of the liquid filter *x(t)* not only depends on the current inputs *u(t)* but also on its past instances or formally [45]:

$$x(t) = (L^M \cdot u)(t) \qquad \textbf{(2-13)}$$

Beyond *x(t)*, observer neurons or readout functions (f^M) can extract the produced mapping information at time instances *(t)* into some output *y(t)* as in Equation (2-14) and also illustrated in Figure 2-10. Readout neurons are where actual classification occurs.

$$y(t) = f^M(x^M(t)) \qquad \textbf{(2-14)}$$

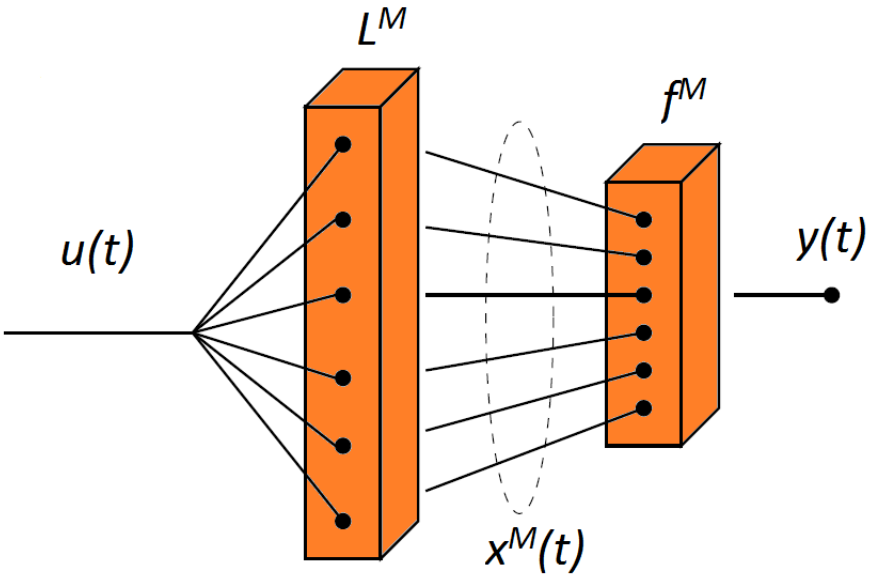
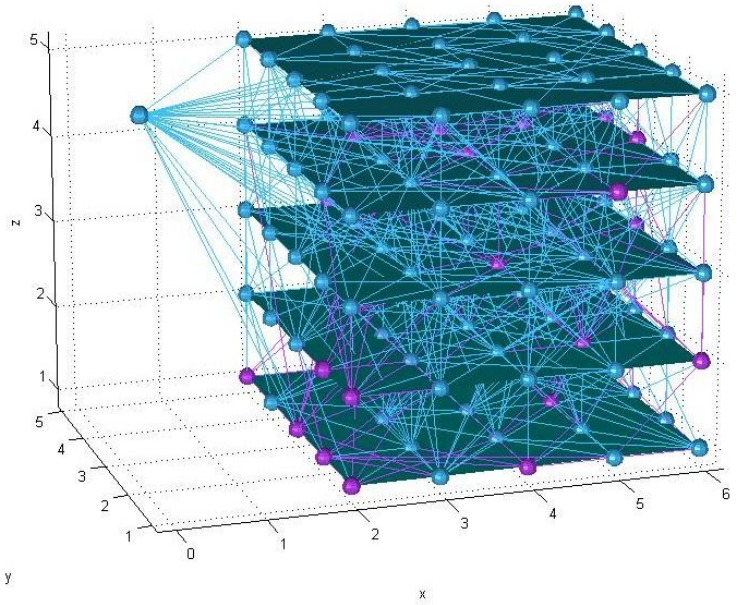
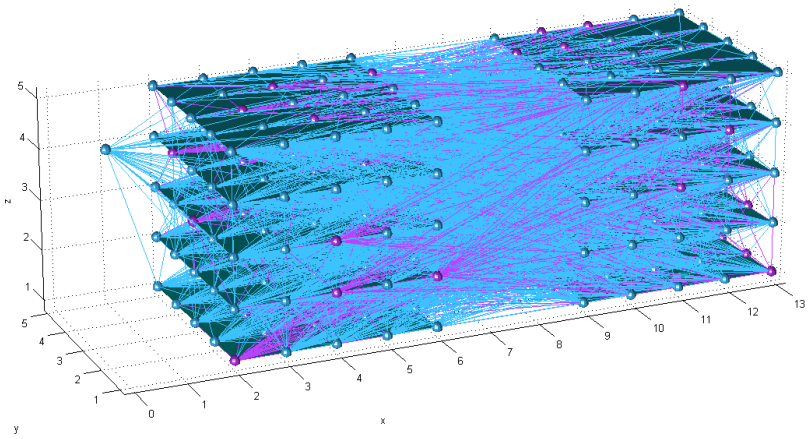


Figure 2-10: Generalised structure of a Liquid State Machine (LSM) [45]**.**

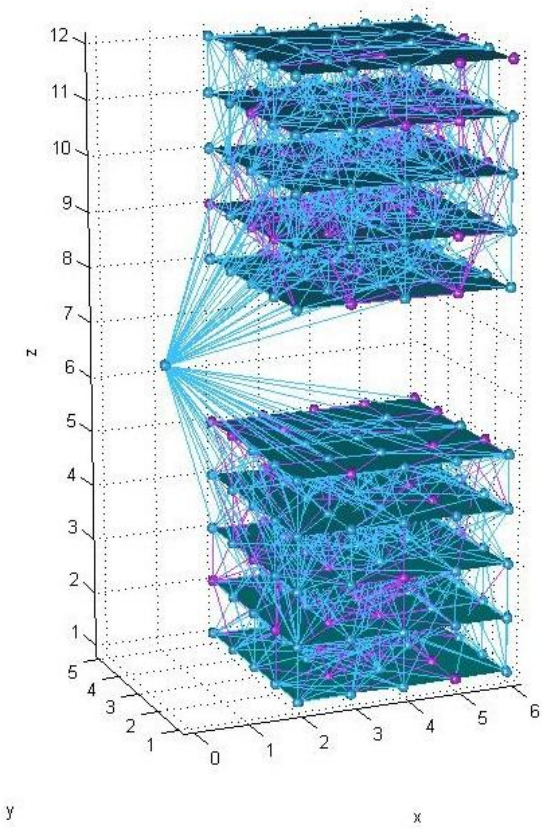
The following Figure 2-11(a) shows a simple architecture created and illustrated with the LSM toolbox [39] in MATLAB. Lines between neurons symbolise the synaptic connections between them. The input neuron (left) feeds a liquid filter (right) structure of 5x5x5 neurons. It is relatively easy to create alternative structures such as in (b) where a single spiking neuron is connected in series to two 5x5x5 liquid filters and in (c) where a single spiking neuron is connected in parallel to two 5x5x5 liquid filters



(a)



(b)



(c)

Figure 2-11: 3D grid examples of LSM layouts in MATLAB.

In addition, LSM have two essential properties [45]:

- Point-wise separation between any two different input functions which ensures that a liquid filter can always separate them.

- Universal approximation under which regardless of the input, a readout function *(f)* is always able to identify the different input functions and convert them into other representations.

LSM being adaptive are not customised for a particular task and can be used as a framework to incorporate all kinds of neural models, synapses and learning mechanisms. Furthermore, LSM perform multiple computations by arranging multiple liquid filters either in parallel or in series. In addition, they accept many simultaneous inputs and via the readout neurons can perform multiclass pattern recognition tasks. In contrast to fixed topology neural networks, the decision only rests on a sufficient liquid filter size which has to be large enough to represent the states of the input data without any further rigorous knowledge.

Although some areas of the brain (or even bacterial life forms) have been identified to work similarly to LSM [46], [47], it is still debateable if LSM can be generalised for the entire brain. The arbitrary nature of recurrent connections within the liquid filter does not provide any insight on the biological functions and structures of the brain. Furthermore, the structure of the framework itself provides little control over processing events within the liquid filter. Finally, LSM do not have standardised mechanisms to optimise their structure against the number of calculations actually required, and only a few studies on LSM optimisation have emerged to address this aspect such as in [48], [49].

2.2.8 Synaptic Plasticity - Learning

Learning in biological terms refers to the change of synaptic efficacies, a process also called synaptic plasticity (section 2.2.5). This learning process regulates the facilitation or depression mechanisms of a synapse, effectively fluctuating the probability of transmission between neurons. In artificial synaptic plasticity, there are two approaches *supervised* in which learning occurs from labelled training samples and *unsupervised* in which training samples are of unknown origin.

Donald Hebb in 1949 had postulated that when an axon from a presynaptic neuron is about to excite the postsynaptic neuron either repeatedly or persistently, then a growth process occurs such that the presynaptic efficiency would increase [21]. Assuming that η is the learning rate then the change of efficacy between the presynaptic (x_i) and postsynaptic neuron (x_j) is:

$$\Delta w_{ij} = \eta \cdot x_i \cdot x_j \qquad \textbf{(2-15)}$$

Equation (2-15) describes synaptic weight facilitation but does not consider synaptic depression or any other realistic mechanisms. It can be expanded by simply adding a decaying term [23]:

$$c = -\eta_1 \cdot w_{ij} \quad \textbf{(2-16)}$$

Additionally, a saturation term can be introduced such that:

$$c_{sat} = \eta_2 (1 - w_{ij}) \quad \textbf{(2-17)}$$

Equation (2-15) can therefore be rewritten by introducing equations (2-16) and (2-17) as:

$$\Delta w_{ij} = c_{sat} \cdot x_i \cdot x_j - c \quad \textbf{(2-18)}$$

In equation (2-18), terms c_{sat} and c ensure that saturation occurs at $w_{ij} = 1$ and decay to $w_{ij} = 0$. Another expansion of equation (2-15) is to include presynaptic or postsynaptic gating [23]:

$$\Delta w_{ij} = \eta \cdot (x_i - v_\theta) \cdot x_j \quad \textbf{(2-19)}$$

$$\Delta w_{ij} = \eta \cdot x_i \cdot (x_j - v_\theta) \quad \textbf{(2-20)}$$

Equations (2-19) and (2-20) are the Hebbian learning presynaptic and postsynaptic gating equations in which v_θ is the threshold voltage. Other variants of Hebb's rule include the covariance matrix, i.e. values of the presynaptic and postsynaptic neuron deviate around mean values, Oja's rule which incorporates quadratic synaptic decay and perhaps the more complex Bienenstock-Cooper-Munroe (BCM) rule by adding a separate nonlinear function [23].

In section 2.2.5, the crucial role of facilitation and depression in dynamic synapses was outlined. Experiments [50], [51], demonstrated that spike timing played a role in these mechanisms, such that when a presynaptic spike would arrive sooner than a postsynaptic activation, then the synaptic weight would increase. Conversely, if a presynaptic spike would appear later than the postsynaptic activation the synapse weight would decrease. In other words, a causal relationship between spikes of a presynaptic and postsynaptic neuron was experimentally found which extended the original Hebbian hypothesis. This process was termed as Spike-Timing-Dependant Plasticity (STDP) and in biology it interestingly has a reverse process called anti-STDP, which has the exact opposite effect [23].

More recently, in Supervised Hebbian Learning (SHL), a “teaching” signal drives the Hebbian learning process so that weights are adjusted by tweaking synaptic currents [52]. Lastly, a supervised STDP variant called Remote Supervision Method (RESUME) [53] integrated SHL with STDP without using current manipulation. In spite of the models explained in this section and the acquired knowledge in the field, it is difficult to identify a standard model which fully captures the brain’s functions and especially human vision.

3 LIGHT PERCEPTION

In this chapter neurons and their functions are examined for visual perception. The journey of light is followed from reception to early processing inside the brain. In section 3.1 the perception of light from the eye and retina is explained. The discovery and understanding of these biological methods have become the reference point for imaging science and along with the nature of the light itself have led directly or indirectly to myriads of applications for cameras, imaging systems and software. The two important phenomena of centre - surround and opponency in the retina are explained in section 3.2, and stemming models based on these phenomena, known as colour constancy algorithms, are outlined in section 3.3. In section 3.4, the functions of the primary visual cortex are highlighted along with the theory behind the fundamental units of visual processing, simple and complex cells.

3.1 Eyesight

Eyes are biology's light receivers and the beginning of sophisticated visual systems that many organisms, including humans, have. Figure 3-1 shows the schematic cross - section of a human eye. Retinal cells are sensitive to light and convert it to electrical signals that neurons transmit to other areas of the brain for processing. Between animal species eyes may exhibit differences, notably the difference between compound insect and crustacean to vertebrate eyes. Specifically, in the human eye the iris, pupil, cornea, ciliary muscle, lens and various surrounding sub-compartments are parts of an optical system solely designed to focus light onto the light sensitive retina. This has to be so precise that minor abnormalities can cause inconsistency to vision, e.g. myopia, astigmatism, presbyopia etc. Other functions such as pupil contraction and expansion regulate the amount of light entering the eyeball from a bright or dark environment.

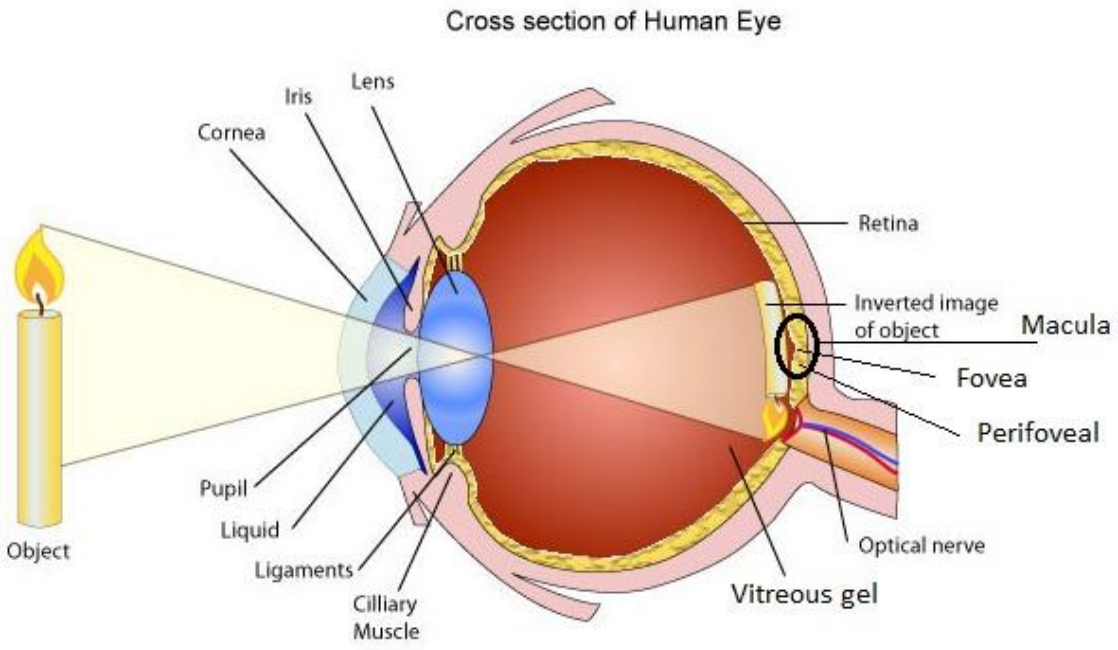


Figure 3-1: The eyeball and its main components [54]**.**

In 1866, Mark Schultze was the first to observe the anatomic structure of the retina under a compound telescope and describe the different types of cells within it [55]. The retina is a layered structure of millions of cells situated at the back of the eye. Its main purpose is to receive the forwarded light from the lens and encode it to pulses. There are five retinal types of cells, receptor, ganglion, bipolar, amacrine and horizontal [56].

Cones, rods and photosensitive retinal ganglion cells (pRGC), are all photoreceptors that excite via an effect called phototransduction. During phototransduction, as soon as a photon impacts the retinal pigment epithelium, photosensitive proteins called opsins begin a chemical chain reaction which either hyperpolarises or depolarises photoreceptors causing the generation of electrical stimuli [9]. Photoreceptors are situated at the bottom layer of the retina. This backwards arrangement may help the pigment epithelium (shown as squares next to the rods and cones in Figure 3-2) to protect subsequent retinal layers from excess light and unwanted scattering phenomena within the eye or may have an additional nutritional significance since the retinal epithelium has also cell nourishment and replenishing responsibilities [56].

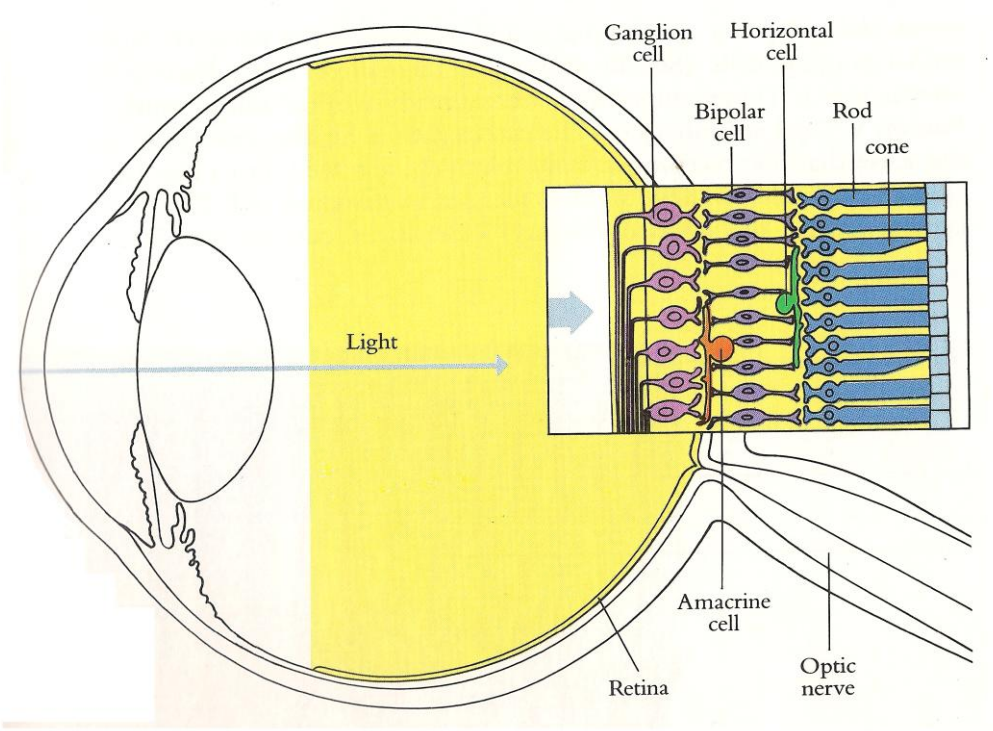


Figure 3-2: As the light arrives at the retina, after passing some translucent membrane layers, it first enters through the ganglion, bipolar, amacrine and horizontal cells to reach the photoreceptors [56]**.**

The number of existing photoreceptors varies from person to person and in the same person from time to time, but on average each eye's bottom retinal layer consists of around 5 million cones, 120 million rods and 100 thousand pRGC [57]. However, it is only cones and rods that contribute to the actual image formation and processing, pRGC are linked with circadian rhythms (the "body clock"), some light normalisation in the retina and suppression of hormones such as melatonin [58]. Cones are photoreceptor cells of conical shape with sensitivity, in the human retina, to three different wavelengths in the visible spectrum of light (as shown in Figure 3-3). Cones are thus sensitive to colour and show fast but otherwise poor adaptation to dark situations, by having low synaptic convergence i.e. their synapses directly feeding bipolar cells (section 2.1), transmit a high amount of visual information [57].

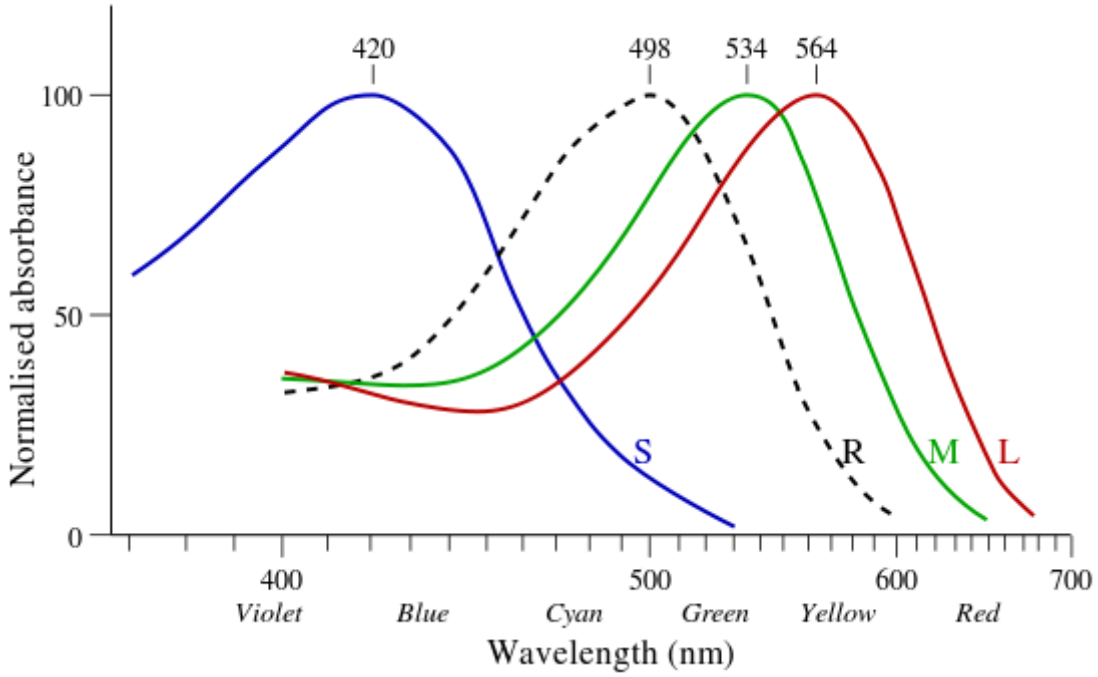


Figure 3-3: The mean absorbance spectra of human photoreceptors. At 420nm peak of blue cones (S for short wave), 498nm peak of rods, 534 nm peak of green cones (M for middle wave), 564nm peak for red cones (L for long wave) [59]**.**

Rods on the other hand, peak only at one wavelength (as shown in Figure 3-3) and although they respond slower compared to cones, they do so more efficiently in the dark. With significantly higher synaptic convergence they tend to provide visual information in larger areas of the visual scene than cones would.

The peripheral retina has a combination of cones and rods with a significantly higher concentration of rods [57]. The fovea centralis is an area of approximately 1% of the total retinal surface, see Figure 3-1. It sends almost 50% of the total visual information down the optic nerves whereas the remaining 50% is transmitted by the perifoveal area and peripheral retina. The central foveal area has a high concentration of compact cones and their existence there leads to sharp central vision in humans [57].

3.2 Centre-Surround and Opponency

Ewald Hering's work on spectral and spatial vision in the 19th century paved the way for better understanding of human visual cognition. His work on biological spectral vision was the first to reveal the centre-surround and opponency mechanisms.

The ganglionic layer hosts retinal ganglion cells whose long axons extend as far as the primary visual cortex to transmit visual information. There are approximately 1-2 million retinal ganglion cells [9] and their distribution across the retina relies on the location of photoreceptors. Ganglion distribution is relative to the size of the receptive field i.e. the volume of a visual scene, of a particular group of photoreceptors. In the peripheral retina, the ganglion cells have inputs of several hundred or thousands of photoreceptors whereas in the foveal area the ratio for cones falls to almost one to one [57], [60]. Furthermore, there are two types of ganglion cells, the on-centre and off-centre which are organised in a structure governed by the intermediate bipolar cells' receptive fields, an arrangement known as "centre-surround".

In the centre region, the corresponding bipolar cells receive stimuli from a small number of photoreceptors as opposed to the surrounding region which is connected to a higher number of photoreceptors. Centre bipolar cells are connected to photoreceptors directly (vertical pathway) while surround bipolar cells are connected via horizontal cells (lateral pathway) [9]. All bipolar cells are connected to either cones or rods but never both. In fact, there are several specialised types of bipolar cells for cones but only one for rods [9]. Centre bipolar cells are directly connected with the subsequent ganglion cells and surround bipolar cells via amacrine cells. Horizontal and amacrine cells combine the surround visual information from photoreceptors and bipolar cells respectively.

Like the ganglion cells, there are two types of bipolar cells, the on-centre and off-centre. An on-centre bipolar cell excites when light is absorbed by receptors at the centre of the receptive field and inhibits when light is absorbed from the surround only. When light is present at both centre and surround regions then in an antagonistic manner they cancel each other out by producing the difference of excitation and inhibition in spikes [56]. This antagonistic behaviour between cells is often termed as spatial opponency or simply opponency. On the other hand, for an off-centre bipolar cell light in its centre causes it to inhibit and excitation occurs when light is present at the surround. When light hits both centre and surround regions opponency takes place again. It is important to note that bipolar and consequently ganglion receptive fields, overlap and so if light shines on one part of the retina the simultaneous activation of thousands of corresponding on-centre and off-centre cells are triggered [61]. Receptive fields are shown for illustration purposes as spatially equal (in Figure 3-4) but in reality receptive fields can vary widely from 0.5 to 2 arcminutes of the total visual field [56].

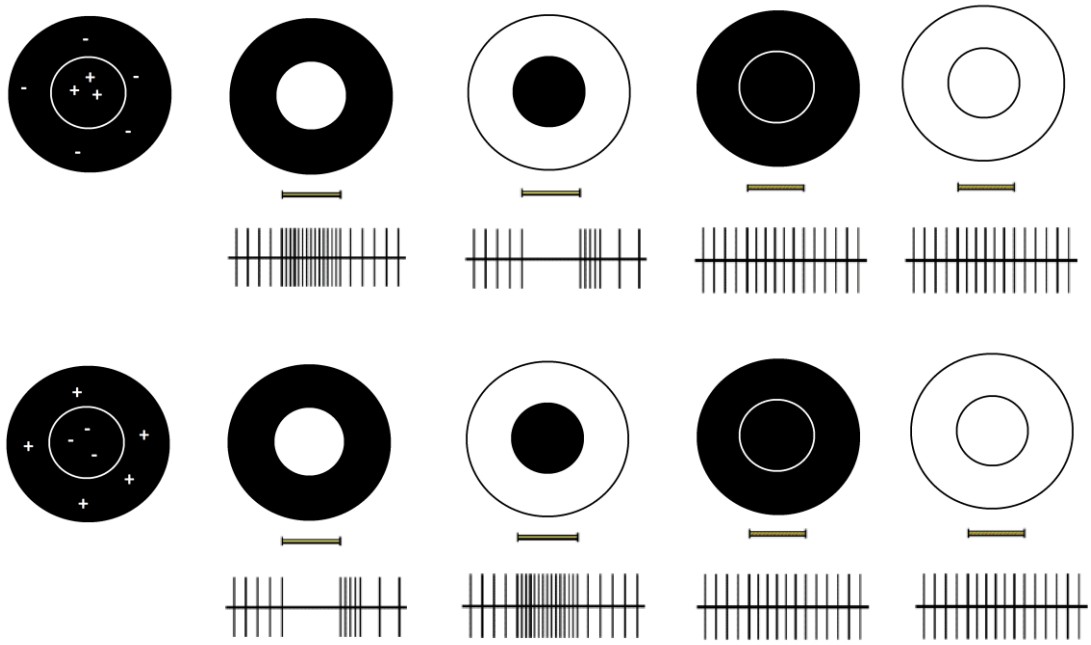


Figure 3-4: Centre-surround receptive fields of on-centre and off-centre bipolar cells. Top row shows an on-centre bipolar cell. It is stimulated (+) when light is absorbed in its centre (shown from the increase in spike production across centre) and inhibited (-) when light hits the surround (shown from the absence in spike production across centre). In absence of light or presence of light on both regions, the cell fires spikes as if by subtracting excitation and inhibition. Bottom row shows an off-centre bipolar cell and the reverse effect.

Retinal on-centre and off-centre receptive fields have been found to be modelled accurately by using the Difference- of-Gaussians (DoG) or “Mexican hat” function [62].

$$f(x) = \frac{1}{\sigma_1\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma_1^{\,2}}\right) - \frac{1}{\sigma_2\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma_2^{\,2}}\right) \quad \textbf{(3-1)}$$

In equation (3-1), σ_1 and σ_2 are the two different standard deviations of the two Gaussians and an illustrative example of this is shown in Figure 3-5.

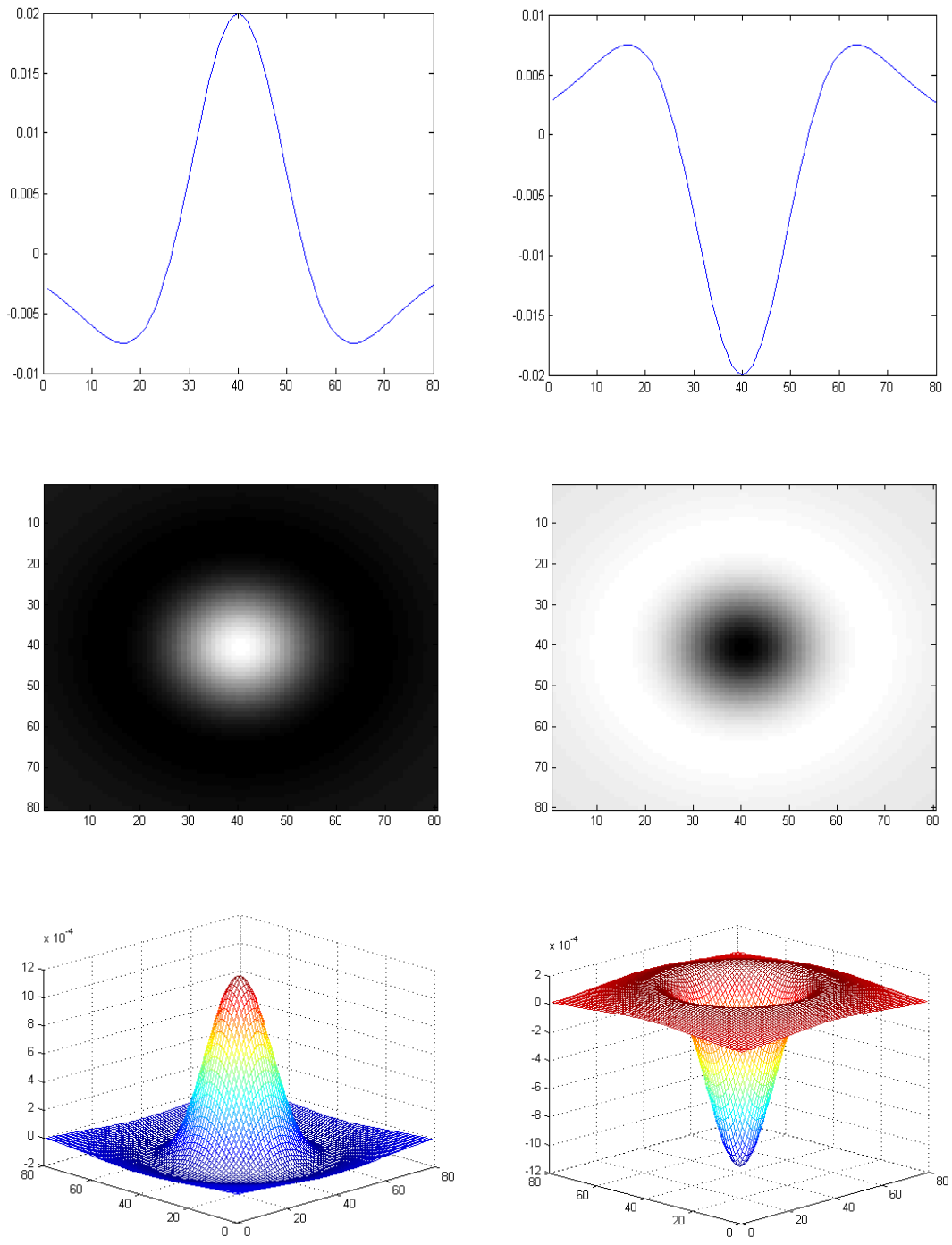


Figure 3-5: 1D, 2D and 3D illustrative examples of on-off DoG receptive fields using MATLAB.

Centre-surround operation in retinal ganglion cells has so far been examined under the presence or absence of light. However, as seen earlier in this chapter (Figure 3-3), cones respond to different wavelengths of light. In this case, it is known [63], that the on-centre and off-centre bipolar cells react in the same centre-surround method under colours red-green and blue-yellow. So for

example, certain on-centre cells sensitive to green are going to respond when green light only hits the centre of their receptive field while they are inhibited when green is at their surrounding region only and vice versa for the off-centre and this is depicted in Figure 3-6. This centre-surround antagonistic process for the retinal cone wavelengths, is called colour-opponency. Equations that describe this colour opponency process are presented in sections 3.3.2 and 4.1.

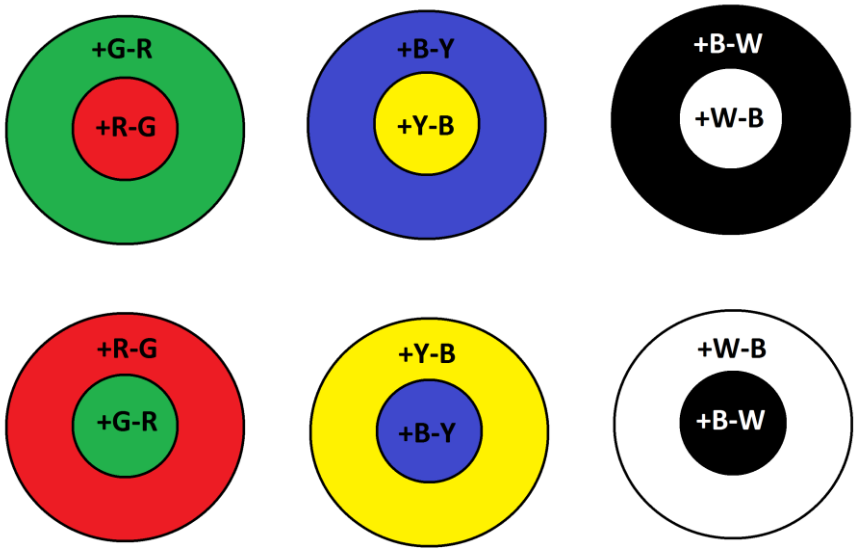


Figure 3-6: Examples of on-centre and off-centre receptive fields in bipolar cells for colour opponency operations. R stands for Red, G for green, Y for yellow, B for middle column is Blue, W for white and B for right column for Black. The plus sign refers to when the particular colour is on and the minus off. Note that if both centre and surround regions contain the same colour, as explained in this section, they cancel each other.

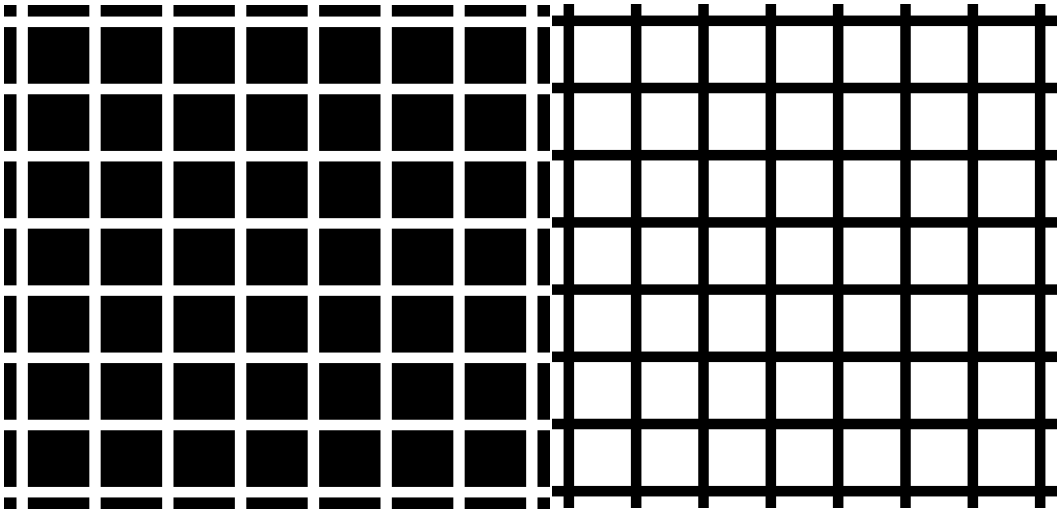


Figure 3-7: The Hermann grid illusion. Grey spots appear and disappear at intersections of the squares. One theory proposed for this phenomenon is lateral inhibition in the retina.

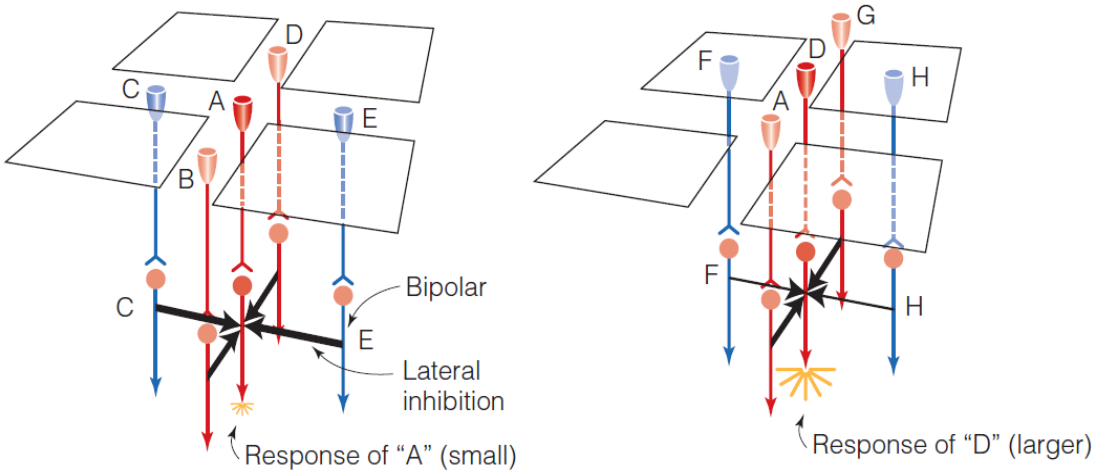


Figure 3-8: 3D view of the Hermann grid showing receptors and their connections leading to lateral inhibition in bipolar cells. A's response at the intersection is smaller than its afferents leading to a grey perception [57]**.**

As seen in section 2.1.1, an important mechanism of neurons is lateral inhibition, a suppressing mechanism that enhances contrast and edges especially in (but not limited to) the retina. In Figure 3-7, if an observer would focus attention at a particular intersection, grey spots appear at others then as attention is shifted towards these spots, they disappear. Five receptors (or groups of receptors) are incident on each intersection and the two corridors around it (Figure 3-8) from the A's perspective. However, bipolar cells B, C, D and E receive a combination of black and white light from their surrounding

receptors (F, H Figure 3-8) thus driving their lateral inhibition upwards. All of the surrounding receptors of A have therefore increased lateral inhibition (contrast enhancement to perceive the borders of the corridors) and as a result push A's behaviour partially, towards grey tones.

3.3 Colour constancy

3.3.1 Overview

In addition to colour opponency, the human retina in conjunction with other parts of the visual system (lateral geniculate nucleus, V1), exhibits a phenomenon called colour constancy where detection of the colour appearance of an object is achieved despite illumination variations of the ambient light [64], [65]. Chromatic adaptation is accomplished in the primary visual cortex by cone cells transmitting the amount of intensity within colours and rods the achromatic intensity of ambient light. This human vision trait portrays the remarkable ability to disassociate dilutions (intensity shifts between black and white) from colour effortlessly, while preserving much of the actual spectral information of an object. This illumination-invariance characteristic is crucial because it makes biological spectral vision dynamic, i.e. performing consistently under a wide range of light changes.

A particular wavelength of the light spectrum depends on [66]:

- The spectral power distribution *E(λ),* which is the amount of energy produced from a light source.
- The spectral reflectance function *L(λ),* which characterises the proportion of light reflected from a surface.
- The light sensor *S(λ),* which determines its sensitivity to that particular wavelength.

$$\rho_k = \int_{\omega} \mathrm{E}(\lambda) L(\lambda) S(\lambda) d\lambda \qquad \textbf{(3-2)}$$

In equation (3-2), ρ is the colour response, ω the visible spectrum, k is the colour channel, assumed here trichromatic (*R, G, B*). It is evident from the equation above that spectral perception for a given surface, depends on the illumination changes of the source as well as the reflectance characteristics of the surface itself. Colour representation cannot be therefore accurate without some form of colour constant perception in the light sensor. Such properties have been proven to exist in the human visual system as mentioned above and accurate artificial spectral representation of surfaces and objects for improved detection, segmentation and recognition algorithms is imperative.

Colour constancy techniques have been developed for more than 30 years. It was Edward H. Land in 1971 [67] that thoroughly examined the existence of this phenomenon and introduced the term “retinex” (Retina – Cortex) [68]. Today, there are numerous colour constancy artificial implementations, as outlined in [69], [70]. Nevertheless, none of the existing colour constancy algorithms can claim universality and adaptation to all environmental situations, since they all have advantages and disadvantages. Colour constancy models are often categorised in the following ways [70]:

- Pre-calibrated
- Grey world and scale by max
- Retinex
- Gamut
- Statistical
- Machine learning

In pre-calibrated approaches, the camera is tuned to reference illumination conditions and images showing changes with respect to the baseline, are transformed either with general transformation techniques [71], [72] or with diagonal matrix transformations[73–76]. Accurate pre-calibration is difficult and the notion of pre-calibration renders these approaches disadvantageous for fast scene estimation.

Grey world [77] and scale by max algorithms [78] are quite similar in that they estimate each individual colour across the image which either averages out to grey or responds maximally. These algorithms are fast but not adaptive and robust.

Retinex techniques [67], [79–81] follow a more biological interpretation by introducing a centre-surround approach. For example, in Single Scale Retinex (SSR) and Multiscale Retinex (MSR) this is done via a Gaussian function [82], [83]. Retinex approaches have the advantage of claiming biological relevance but can be difficult to tune properly[70].

In Gamut colour constancy methods [74], [84], the reflectance and the illuminant of the scene are assumed to have certain constraints. Moreover, in Gamut methods a baseline reflectance gamut map is created under a canonical illuminant. Similarly, another gamut map is extracted for the image under unknown illumination changes. The image gamut map is projected onto the canonical in an attempt for the two to be matched via heuristic methods. Gamut approaches are computationally impractical and relative to the sensor for canonical illumination estimation.

Statistical approaches take a more mathematical approach. In [85], global scene illumination parameters are estimated using maximum likelihood and Kullback Leibler (KL) - divergence. Using Bayesian theory [86] estimations are made for both the unknown illuminant and unknown spectral information of objects from sensor responses. In contrast, a priori assumptions on known or unknown illuminants are examined as uniform in [87]. Moreover, Finlayson et al [88], employed a technique known as colour by correlation in which the illuminant chromaticity is approximated using correlation. Statistical methods are adaptive and illumination knowledge is not necessary but can also be computationally expensive and impractical for real time applications.

Finally, machine learning methods have two stages, training and testing. Any predictions on testing data solely rely on the accuracy of the training process. The early machine learning methods are applied using neural networks e.g. a multilayer perceptron [89], [90] which show better performance than the colour by correlation method. A parallel algorithm is proposed in [91] and a multiple illuminant hypothesis is examined in [92]. A direct implementation and comparison of a neural network with the human visual system is made in [93] concluding to the hypothesis that there must be biological mechanisms which isolate background colours from foreground, and estimate differences between an object and its background. Other machine learning techniques involve the use of a fuzzy based approach in [94] to complex objects under varying illuminants and in [95], support vector regression is being used to specify illumination chromaticity histograms. Overall, machine learning methods are adaptive and robust although their training step requires time and accuracy.

3.3.2 Fusing colour constancy algorithms

As seen in the previous section of this chapter, a wealth of colour constancy algorithms and methods has been created over the last 30 years. None of the methods however, can claim universal surface colour estimation properties with optimum performance. Under varying illumination conditions, object characteristics and environmental situations, all colour constancy methods behave inconsistently. Therefore in engineering terms, it is more appropriate to find a compromise between them, an algorithm which fuses colour constancy algorithms or some of their aspects for optimum performance. In direct competition with each other, given a particular scene, the winning method should apply colour constancy. The first attempt on this idea was made by Cardei in [89]. Recently such an optimisation approach was examined in [96] which is briefly explained in this section and additional experimentation can be found in section 6.1.2.2.

There have been several approaches to statistical image semantic information i.e. finding the theme of the image holistically and extracting low-level information such as the amount of spectral distribution and the nature of spatial information of a scene [97], [98]. From a biologically-inspired perspective, there have not been any studies on this particular topic. Semantic information could be retrieved with visual attention (section 4.1) techniques, but would require some form of training. Moreover, salient features would provide little information on the useful low-level characteristics of the scene collectively. It seems that if a biological mechanism is specialised for such semantic information retrieval, it precedes visual attention.

Textured responses from visual scenes resemble the Weibull distribution [99]. The Weibull distribution is given by [96]:

$$f(x) = C \exp\left(-\frac{1}{\gamma} \left| \frac{x}{\beta} \right|^{\gamma} \right) \qquad \textbf{(3-3)}$$

In equation (3-3), C is a normalising constant, x is the edge responses of a particular spectral channel (or intensity) to a spatial edge detection filter, β is the scale parameter of the distribution and γ is the shape parameter of the distribution. Parameters β and γ represent the width (contrast) and height (texture coarseness) of the distribution. Thus, β and γ can capture the image's contrast and texture collectively. Detailed experiments of Weibull parameter fitting can be found in both [96], [99].

The capacity of the Weibull distribution to describe texture is introduced for the first time with Gabor filters and biological-like object recognition in section 7.2. The Weibull distribution parameters can be used in conjunction with colour constancy algorithms since there is a correlation between the edges present in a visual scene and the performance of a colour constancy algorithm [96]. More specifically, the algorithm in [96] performs the following steps during training:

1. Convert images to colour-opponent space.
2. Find Weibull parameters β and γ for all images of the training dataset under the same Gaussian derivative edge filter.
3. Given these values of β and γ, apply colour constancy algorithms and estimate illuminants.
4. Compute the median error of all colour constancy algorithms.

5. Use a Support Vector Machine (SVM) to learn which colour constancy algorithm has the lowest error and therefore best describes a specific set of values β and γ in images.

At the testing stage the SVM is used on the extracted Weibull parameters of an unknown image under the same Gaussian filter. The best colour constancy algorithm is chosen to represent the unknown testing image. The opponent space in step 1 is given by [96]:

$$O_1 = \frac{R-G}{\sqrt{2}} \quad \textbf{(3-4)}$$

$$O_2 = \frac{R+G-2B}{\sqrt{6}} \quad \textbf{(3-5)}$$

$$O_3 = \frac{R+G+B}{\sqrt{3}} \quad \textbf{(3-6)}$$

In equations (3-4), (3-5), (3-6) *R*, *G* and *B* are the red, green and blue bands of an image. O_1 is the Red-Green opponency, O_2 is the Blue-Yellow and O_3 is the average of all bands i.e. the intensity. These equations are also examined in section 4.1.

Furthermore, step 4 of the algorithmic procedure above relies on the error measure between algorithms and is given from the angular error given as [96]:

$$e_{tot} = \cos^{-1}\left(\frac{e_l \cdot e_e}{\|e_l\| \cdot \|e_e\|}\right) \quad \textbf{(3-7)}$$

Equation (3-7) shows, the dot product between the actual illuminant (e_l), obtained from ground truth data and the estimated illuminant (e_e) extracted from the colour constancy method over their Euclidean norm. A thorough investigation of this algorithm followed by experimentation is given in section 7.1.2.

3.4 Primary Visual Cortex

Ganglion axons carry visual signals through the optic nerves from both eyes which cross at the optic chiasm, situated at the base of the hypothalamus. After the optic chiasm, in which the optic fibres split, the signals reach the lateral geniculate nucleus (LGN) and the superior colliculus (SC) as depicted in Figure 3-9.

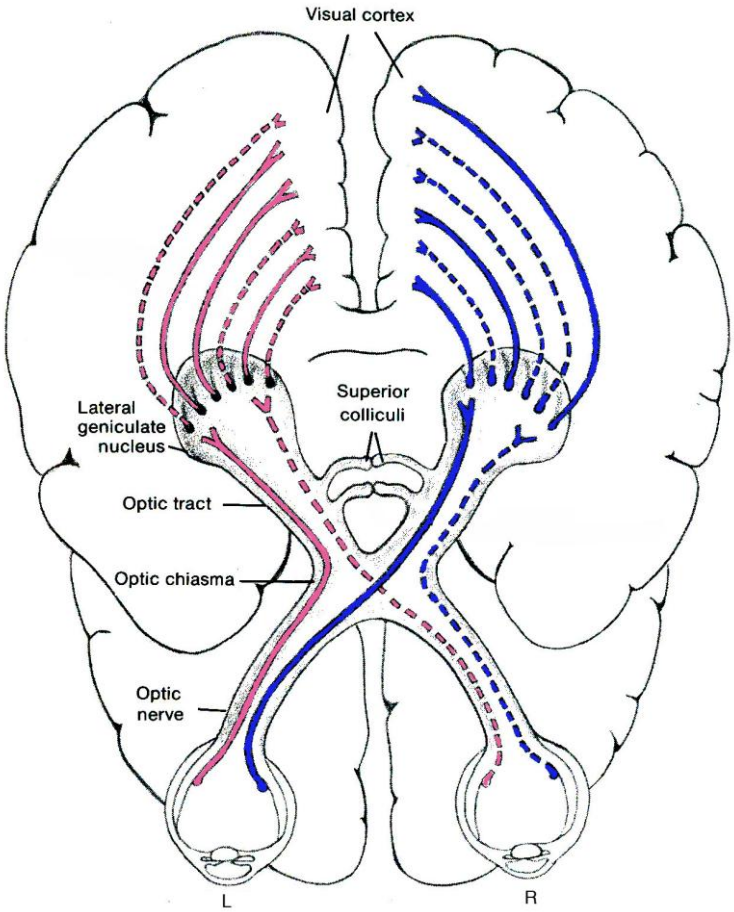


Figure 3-9: The visual system extending from the eyes to the primary visual cortex [57]**.**

The SC, through a combination of audiovisual information, helps a person to orient their head and eyes towards a source of interest. Similarly, the LGN plays a pivotal role in rapid eye movements towards the direction of visual attention and spatiotemporal decorrelation. It consists of six layers (1-6) that receive the visual data from the ganglion cells' axons as mentioned above. These layers' main role is to separate and categorise the stimuli received from the eyes in a very basic level. Layers 1 and 2, consist of magnocellular cells (M cells) which are connected to the ganglion cells of rods and are responsible for perceiving form, movement, depth and variance in brightness [100]. Layers 3, 4, 5 and 6, are collections of parvocellular cells (P cells) accepting input from the ganglion cells of cones and thus aid in perceiving colour from medium-long wavelengths (the green-red regions) and detailed edges [101]. In between all the layers, Koniocellular cells (K cells) are connected to cones only sensitive to short wavelengths (blue region) and are thought to compensate for the perception of trichromatic vision [102].

It is through the LGN that optic axons of relay neurons transfer the information to the visual cortex. At that part of the brain, these collections of axons are known as optic radiations or geniculo-calcarine tracts. These tracts are the last section of the pathway that visual information travels through to reach the visual or striate cortex (Figure 3-10).

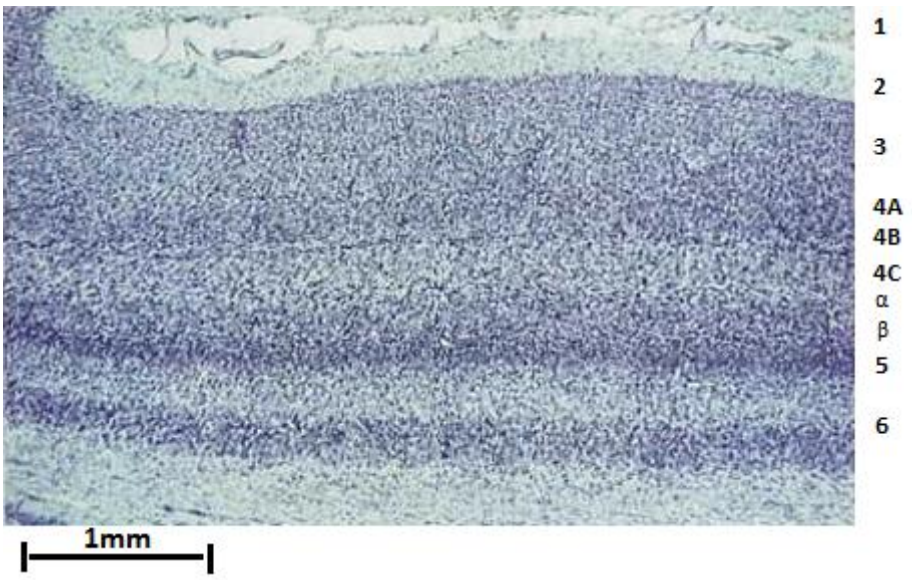


Figure 3-10: Cross-section photograph of the striate cortex. At this spatial resolution, different cell strata are just becoming visible [56]**.**

Visual information from parvocellular and magnocellular cells first reaches layer 4C (regions α and β) of centre-surround simple cells and then propagates backwards through layers 3, 2, 1 then 5 and 6 [56]. From layer 6 it then progresses to deeper layers of the brain. Layers in the visual cortex are alternating between simple and complex cells, explained in more detail below.

3.4.1 Simple and complex cells

Simple cells like retinal ganglion cells have on and off regions in which they show excitation and inhibition. In contrast to ganglion cells, simple cells do not share the same centre-surround circular symmetry. Their receptive fields rather resemble rectangular shapes as defined by straight bars in their centre enveloped by two parallel bars on either side in their surround region. Similarly to retinal cells they operate in an opponent fashion [56], as shown in Figure 3-11 below.

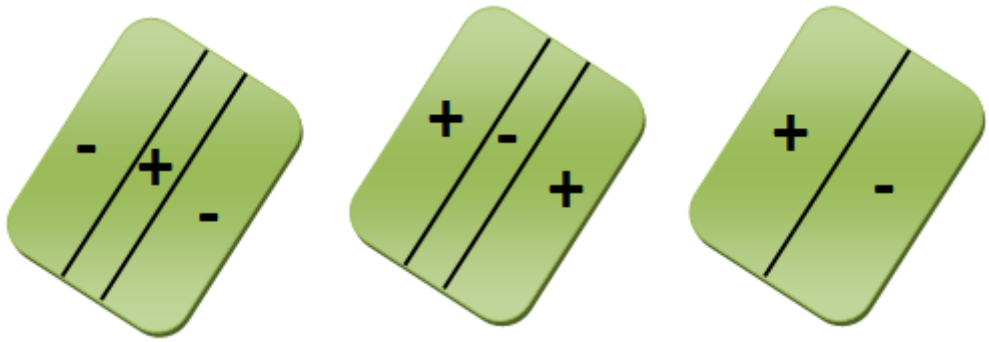


Figure 3-11: Simple cell centre surround rectangular receptive fields illustrative examples. First from the left shows an on-centre simple cell, middle receptive field is for an off-centre simple cell and third shows an edge receptive field without distinct centre surround regions.

Simple cell receptive fields vary in size according to their position. This follows from the fact that their inputs are via LGN, the retinal ganglion cells and their topology directly influences simple cells. Simple cells have receptive fields from 1-2 arcminutes to 1 degree centre regions [56].

Figure 3-12: Examples of edges appearing in the receptive fields of on-centre simple cells. The left receptive field would produce the maximum response while a smaller one would be observed for the middle. In the right receptive field there would not be any response.

In Figure 3-12, in the left hand figure if a bar has appeared at the optimum position then the simple cell would fire rapidly. In the middle figure, if a bar appeared in the inhibitory area then an off discharge would be released. Lastly, in the third situation if it hits both centre and surround regions then the cell does not respond at all.

Complex cells on the contrary, do not have centre-surround but overlapping on and off regions [103]. They tend to have larger and more circular receptive fields compared to simple cells since often they receive their inputs from

multiple simple cells. Two thirds of the striate cortex cells have been found to be complex and thus substantially outnumber simple cells. Complex cells emit responses regardless of where an edge appears in their receptive field. They are orientation-selective i.e. an edge must be aligned in a specific orientation for the complex cell to respond. Moreover, about 10-20% of complex cells respond only when a bar moves along a specific direction as depicted in Figure 3-13 [56].

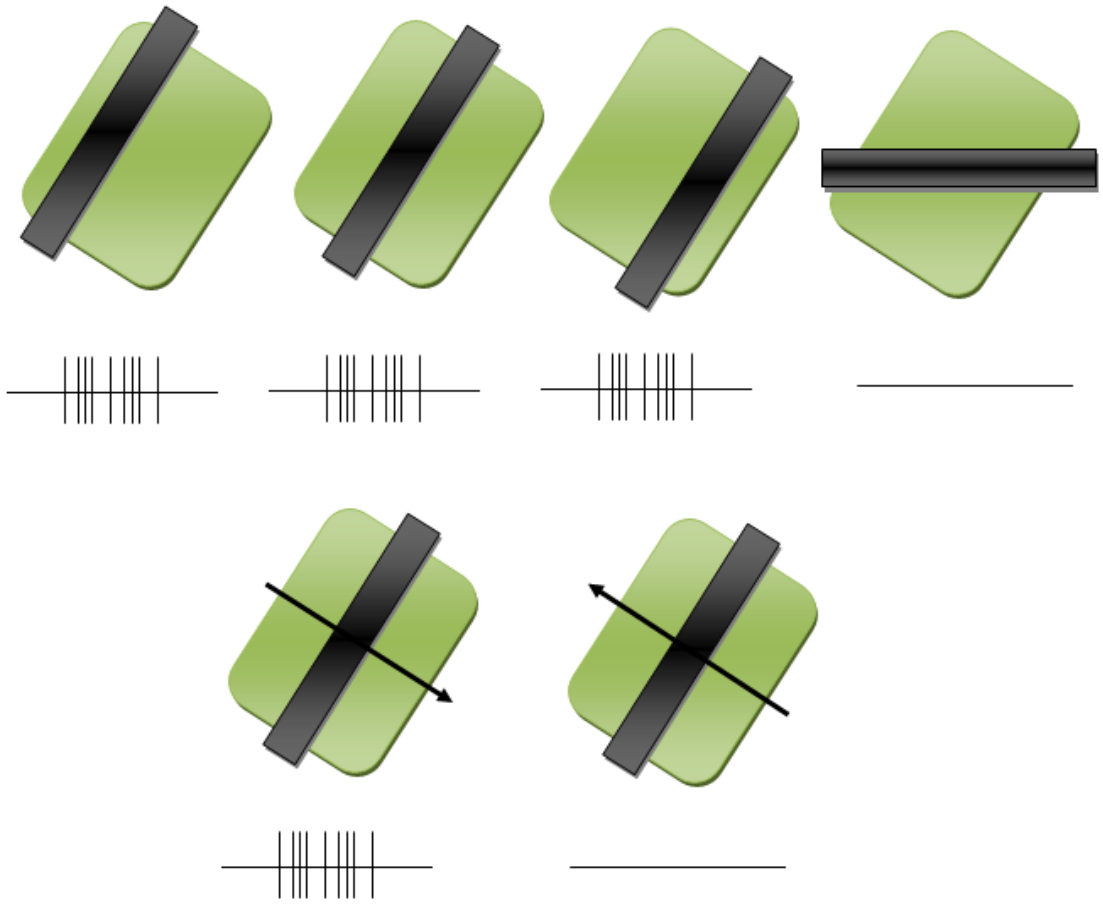


Figure 3-13: Complex cell receptive fields illustrative examples. Top row from left shows a bar being moved across the orientation- selective complex cell receptive field. In the last case, if a bar is aligned in a different orientation then no response would take place. Bottom row shows an example of the direction selectivity property that some complex cells also exhibit [56]**.**

Finally, there is a subcategory of complex cells called hypercomplex or end-stopped cells. These respond maximally to edges of a particular length moving along a particular direction[57], [104].

3.4.2 Simulation of simple cells via Gabor filters

Simple cells are accurately modelled using two dimensional Gabor filters [105–107]. A Gabor filter is a linear filter defined as the product of a complex sinusoid (known as the carrier) with a 2D Gaussian envelope. The relationship of a Gabor filter is given by:

$$G(x, y) = s(x, y)w_r(x, y) \quad \textbf{(3-8)}$$

In equation (3-8), *s(x,y)* is the carrier and $w_r(x,y)$ is the envelope. The complex sinusoid *s(x,y)* is given by:

$$s(x, y) = \exp\left(i\left(2\pi\frac{x'}{\lambda} + \phi\right)\right) \quad \textbf{(3-9)}$$

While the Gaussian-shaped function by:

$$w_r(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \quad \textbf{(3-10)}$$

Hence, from equations (3-8), (3-9) and (3-10) the relationship for a Gabor filter can be obtained.

$$G(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda} + \phi\right) \quad \textbf{(3-11)}$$

where

$$x' = x\cos\theta + y\sin\theta \quad \textbf{(3-12)}$$

$$y' = -x\sin\theta + y\cos\theta \quad \textbf{(3-13)}$$

In equation (3-11), the λ parameter is known as the wavelength of the cosine factor. Its value is usually in pixels and consequently it is given in real numbers greater or equal to 2. In conjunction, with parameter σ (the effective width), the wavelength specifies the tuning accuracy of the Gabor filter. For example, the following Figure 3-14, illustrates Gabor filters with wavelength parameters of 5, 10 and 15.

Figure 3-14: Varying the wavelength parameter in a Gabor filter (from left to right, parameters 5, 10, 15)

The effective width σ over the wavelength, equation (3-14) below, gives the standard deviation of the Gaussian envelope in the Gabor filter [108]:

$$\frac{\sigma}{\lambda} = \frac{1}{\pi}\sqrt{\frac{\ln 2}{2}} \cdot \frac{2^b + 1}{2^b - 1} \qquad \textbf{(3-14)}$$

In (3-14), b is the half-response spatial frequency bandwidth (in octaves) and is obtained from [108]:

$$b = \log_2 \frac{\frac{\sigma}{\lambda}\pi + \sqrt{\frac{\ln 2}{2}}}{\frac{\sigma}{\lambda}\pi - \sqrt{\frac{\ln 2}{2}}} \qquad \textbf{(3-15)}$$

From equations (3-14) and (3-15), it is evident that the effective width and wavelength are intertwined in such a way that the bandwidth of the Gabor filter is affected by both. Bandwidth is inversely proportional to σ and therefore the smaller the bandwidth, the larger the σ. An example of varying the bandwidth is given below in Figure 3-15.

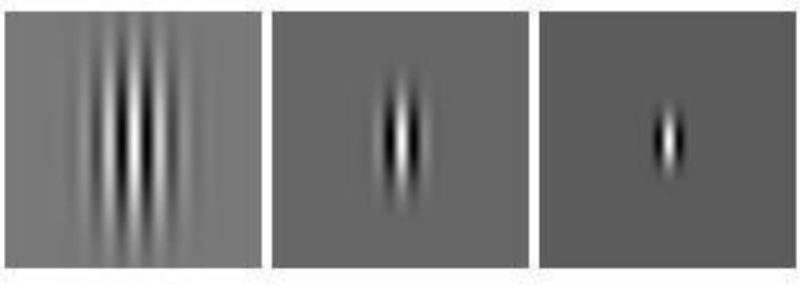


Figure 3-15: Varying the bandwidth parameter in a Gabor filter (from left to right, parameters are 0.5, 1 and 2, where wavelength is constant at 10)

Parameter γ is known as the spatial aspect ratio which typically controls the ellipticity of the Gabor function. When $\gamma = 1$ then the filter is completely circular while for $\gamma < 1$ the Gabor function is elliptical, stretched towards the parallel filter lines (as shown in Figure 3-16 below). Parameter γ has been found not to have a significant impact on edge detection and a default value is usually at 0.5.

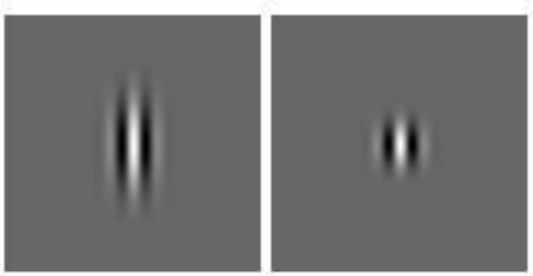


Figure 3-16: Varying the aspect ratio parameter in a Gabor filter (from left to right parameter values are 0.5 and 1)

Parameter θ controls the preferred spatial orientations of the Gabor filter function and values vary in degrees from 0-360 and this is depicted in Figure 3-17.

Figure 3-17: Using four orientations in a Gabor filter (from left to right, 0, 45, 90, 135 degrees)

Finally, the phase offset φ, specified in degrees, controls the cosine function and determines the symmetry of the Gabor filter with respect to its centre and is shown in Figure 3-18 below.

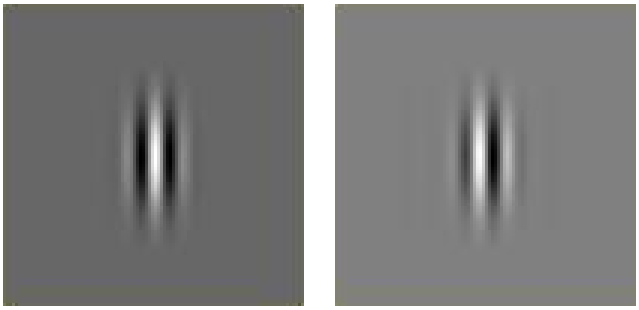


Figure 3-18: Varying the phase offset in a Gabor filter (left image at phase 0 and right at phase 90)

According to [109], the experimental properties which have been derived for V1 cells are:

1. The orientation selectivity differences between cells can be very large, from as narrow as 8 degrees to almost no orientation selectivity at all.
2. The bulk of simple cells have spatial frequency bandwidths from 1 to 1.5 octaves (with an average closer to 1.4 octaves).
3. Orientation selective simple cells exhibit minimum response at about 30-40 degrees away from their optimally tuned orientation.
4. Spiking rates of excited simple cells are between 0Hz to 120Hz.
5. Frequencies vary from 0.5 cycles per degree of visual angle to 15 cycles per degree. Specifically, foveal cells on average (i.e. simple cells that handle information from the retinal ganglion cells connected to cones of the fovea) portray 4.25 cycles per degree where parafoveal cells on average portray 2.7 cycles per degree.
6. It has been found that the fovea is more oriented vertically and horizontally than diagonally.

In Figure 3-19 below the resulting image contains the sum of all orientations at fine detail and Figure 3-20 illustrates the effect parameters have on the coarseness of the edges detected. It is easy to see in Figure 3-20 as either the bandwidth or the wavelength increase so does the coarseness, omitting a large part of the texture information within the zebra's snout.

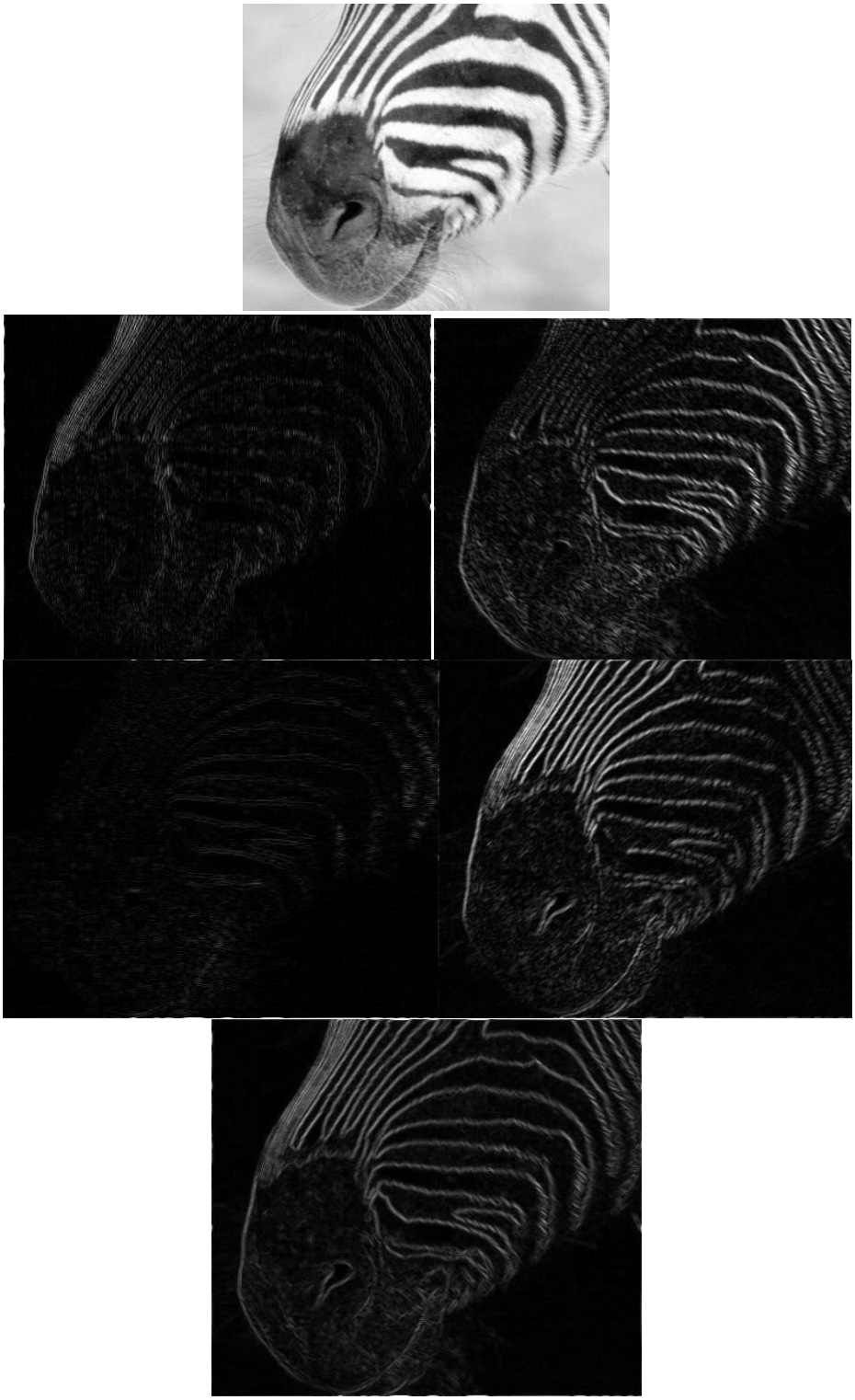


Figure 3-19: An example implementation of a Gabor filter at fine detail (γ=0.3, λ=2, b=3, φ=0, 90). Top input image of a zebra's nose is analysed in four orientations (left to right, 0, 45, 90, 135) and the resulting image at the bottom contains all orientations.

Figure 3-20: Varying the wavelength and bandwidth parameters to create coarser details (i.e. shape information) using all four orientations. Middle row on the left γ=0.3, λ=8, b=5, φ=0 and 90, middle row on the right γ=0.3, λ=8, b=2, φ=0 and 90, bottow row γ=0.3, λ=12, b=2, φ=0 and 90.

Knowledge of the tuning properties of V1 simple neurons is helpful in neuroscience and its biological-like vision algorithms since the V1 area is the earliest and most fundamental image processing centre. Consequently, all of the algorithms described in the following sections have the V1 feature extraction theory in common and utilise it to extract their spatial features.

Gaussian derivatives have also been proposed and implemented as alternatives to Gabor filters [110], [111] but Gabor filters possess more parameters thus allowing accurate manipulation and optimisation of the receptive field.

3.4.3 Simulating complex cells

Construction of complex cells, unlike the use of linear filters in simple cells examined in the previous section, is not straightforward in that it must exhibit the combined properties described in section 3.4.1. It has also become recently evident that complex cells may play a role in a variety of biological image processing processes which alter their behaviour [112]. More than 60 years after their discovery, complex cells are still a topic of active research and theoretical models are still elusive.

There have been several approaches developed to approximate their behaviour explicitly. Some studies concentrate on the representation of the complex cell receptive fields while others focus on their properties. Authors in [113] have employed Gabor magnitudes, in [114] a formulation based on DoG and more recently in [115] Gaussian derivatives to model complex cell receptive field behaviour following biological experiments from Hubel and Wiesel [116]. In all these implementations however, there is no experimental evidence to account for the invariance and generalisation properties of complex cells.

On the other hand, Riesenhuber and Poggio in [117] proposed a hierarchical approach using a “MAX-pooling” mechanism for complex cells based on psychological and physiological evidence such as in [118–120]. This is a more general approach aimed at achieving position and scale invariance rather than a physiologically faithful complex cell receptive field. Under this approach the strongest of the afferent simple cells supplies the complex cell. This max-like mechanism is examined and explained in more detail in section 4.2.3.

4 VISUAL SCENE INTERPETATION

In image processing, biological visual systems have been a topic of extensive research for many years. Since the Nobel Prize winning research on mammalian visual perception by David Hubel and Torsten Wiesel [116] was originally introduced in 1962, many studies sprang up to create the foundations of biological-like machine vision. Extensive research [121–124] had further revealed the workings of a fundamental “two cortical streams” function of the primates’ visual cortex. Generally, the dorsal visual cortical stream or “where/how pathway” is responsible for visual attention i.e. detection of regions of interest within the visual scene and the ventral stream or “what pathway” which executes object recognition and establishes associations within the scene and particularly the regions of interests (see Figure 4-1).

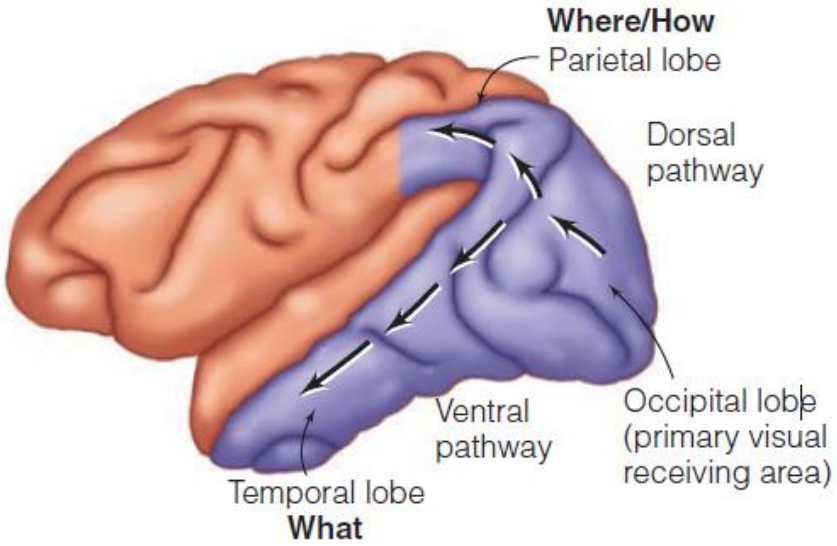


Figure 4-1: An illustration of the two visual pathways in the brain [57]**.**

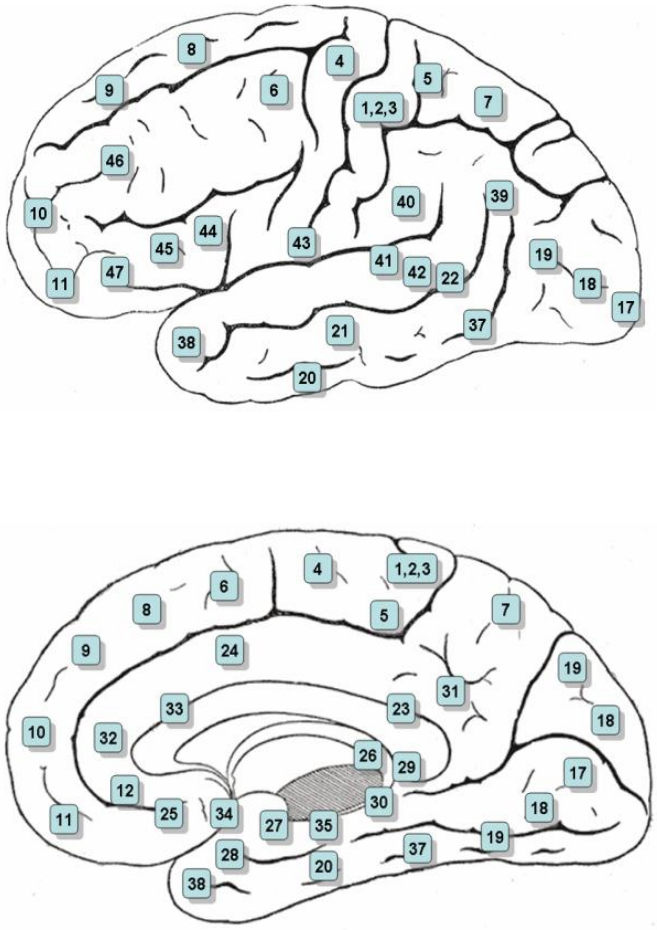


Figure 4-2: The Brodmann areas, the top figure is the lateral surface, the middle figure is the medial surface of the brain, and the bottom figure shows all the visual areas[116], [125], [126]**.**

Today, there is still uncertainty and active research on the precise mechanisms of biological vision in cortical layers after V1. In sections 4.1 and 4.2 outlines of the biological vision processes in the dorsal and ventral stream respectively, are provided. These outlines are followed by a theoretical analysis of the dominant models of visual attention and object recognition which have been the basis of this work.

4.1 Dorsal Stream and Visual attention

4.1.1 Biological Overview

After the V1 region of the primary visual cortex [124], visual data enters the V2 area which is also known as the prestriate cortex. V2 is multilayered and its neurons have broader receptive fields bearing a resemblance to V1 simple and complex cells. Receptive fields combine from both hemispheres of the brain and as a result they decode the depth order of objects (stereoscopic map), distinguishing foreground from background objects [127]. In addition, it has been shown that cells in the V2 respond on average to more complex spatial patterns and contours while playing a role in objection-recognition memory [128].

Subsequent visual area V3 has two sub-regions even though its exact boundaries are still under dispute. The dorsal V3 and the ventral V3 (or VP), play an intermediate and routing role for subsequent visual areas and have broad receptive fields with weak axonal connections to previous regions. V3 contributes in associating all the early features together e.g. global motion with colour [129], [130]. Region V4 has been primarily linked with the ventral stream but has also been proven to contribute in the dorsal pathway [131].

The next visual station of the dorsal stream or “where/how” pathway is visual area V5 or the middle temporal visual area (MT), an extrastiate cortical layer. It has hierarchical feedforward connections from V1, V2, dorsal V3 and koniocellular cells from the LGN [132]. Research shows that this area perceives global motion from complex objects through direction-sensitive cells only and helps in eye movements accordingly i.e. tracking [133], [134]. Area V6 is another key module of the dorsal stream adjacent to V3 receiving inputs from V1 directly. V6 is retinotopically-organised (retinal cell coordinates in the visual field have been preserved) and has been shown to play a role in visuotopic organisation, topography, detection of movement and self-positioning activities [135], [136]. The last extrastriate cortical area V7 on the dorsal pathway is largely unexplored. However, two recent studies [137], [138] suggest possible functions in identifying global symmetry and disparities in motion.

After visual data has been processed by the dorsal extrastriate area V7, it passes into the posterior parietal lobe (Brodmann areas 5 and 7 in Figure 4-2) and the inferior parietal lobe (Brodmann areas 39 and 40). Generally, the parietal lobe has two main functions, it combines all sensory data in a unified cognition system and it reconstructs spatial visual attention data as a coordinate system. These functions assist humans in understanding where objects exist in space and how an action (i.e. eye movements, reaching, walking etc) can be performed in the same space. The parietal lobe has an incredibly large number of neuronal connections with cortical areas of the ventral stream and other parts of the brain.

4.1.2 Visual attention models

4.1.2.1 Biological motivation

Human visual attention is selective since it does not extend beyond a certain area in visual angles, only attending one area of the visual field at a time. According to [139], attention limitations occur because of the processing load and resources that parallel attention may impose on the brain. However, human survival in nature and thus evolution, may have also contributed to shape this process.

There are three categories of human visual attention that have been identified [139]:

1. Spatial Attention i.e. location driven attention, further expressed in two forms, overt attention that focuses on a particular stimulus present in the visual field and covert attention, a process of attending to areas after mental associations and deductions or attending to objects on the periphery of the visual field.
2. Object-Based Attention (OBA), an object's structure driving a visual search approach.
3. Feature-based attention (FBA), an approach based by visual features such as colour, orientation, intensity, motion etc.

Two proposed neural mechanisms are affected by attention, gain modulation and tuning, which are not necessarily exclusive and independent [140], [141]. Gain modulation ensures that neurons respond to a stimulus by a multiplicative factor or attended features get promoted with respect to other non-prominent features (saliency). Tuning, suppresses irrelevant features without increasing responses to attended features (lateral inhibition).

In a recent study, the time course of spatial attention was directly compared to FBA, finding spatial attention faster by almost 200ms [142]. Other studies have

additionally revealed that spatial attention and FBA are activated by different neural mechanisms in the brain [131], [143]. Visual search can also be influenced by objects only [144]. The precise relationship between the three attention categories remains uncertain and neurophysiological evidence shows that if such a relationship exists, it is task-dependant [139], [144]. Regardless, spatial attention and OBA are generally accepted as top-down tasks, driven by intentional influences and FBA as a bottom-up data driven method guided by unintentional biases[145]. For detailed analyses on the three visual attention categories refer to [139], [144]. In this thesis, top-down (intentional visual search) tasks are not taken into consideration.

4.1.2.2 Computational Models for visual attention

Computational visual attention models can be separated in the following broad categories:

- Temporal Tagging
- Emergent Attention
- Selective Routing or Tuning
- Salience Map

In temporal tagging models, excitatory and inhibitory integrate-and-fire neurons compete to make decisions based on differential relationships that control visual attention. Representative examples under this category are [146–148].

Emergent Attention models have a more holistic approach by allowing top-down biases to influence feature extraction on the attention regions. These models resemble saliency maps conceptually and examples can be found in [149–151].

Under Selective Tuning, algorithms are organised in feedforward and feedback networks of units to resemble axonal pathways and modules of the brain. Their configuration is changed so that the optimum representation fits a certain top-down bias, attempting to solve the problem of stimulus choice via a winner-take-all strategy. Some models of this category are [152–154].

Computational attention models are based on psychological and neuro-physiological data but it is difficult to guarantee biological-like performance and consistency. Firstly, there is lack of conclusive physiological evidence on the underlining biological processes and secondly, there is need for a performance measure. Visual attention models in the past have been compared against eye movement data but that cannot account as a realistic benchmark comparison due to the subjective nature of vision. Moreover, the unknown degree of influences in visual attention makes the identification of a standard comparison even harder.

Unconstrained by performance criteria in this work, saliency maps are chosen as the framework of visual attention due to their ease of use, computational efficiency and simplicity, unbiased and bottom-up hierarchical architecture which fits the overall objective of the thesis.

4.1.2.3 Saliency Maps

An early bottom-up and influential FBA study is the feature integration theory [155]. According to this theory, which closely follows the findings in [116], within the visual cortex exist primitive features which are responsible for the creation of feature maps that later formulate a complete saliency map of a visual scene. In this model, saliency maps i.e. 2D topographic collective maps of all conspicuous objects or locations across a scene were introduced. Features of colour, orientation and intensity of neurons compete to give a winning location of salience via centre-surround operations. Moreover, it was suggested that this saliency map exists in the primary visual cortex (V1) of the brain.

The bottom-up concept of the feature integration theory was greatly refined in [156], along with the aid of two previous models [157], [158] to a more elaborate structure of extracting features often abbreviated as the Itti/Koch/Niebur (IKN) model. Since IKN's appearance a number of visual attention models emerged such as [159–164]. These approaches share similarities and differences, in the way that saliency of an image is defined or how it is calculated.

For example, using a Bayesian approach in [159], contextual information is created based on the spatial localisation of objects e.g. a computer mouse is usually close to a computer or clouds tend to be at the top of an image etc. In contrast, in [160] the Shannon information of an event is calculated to find the saliency and histograms are calculated over small portions of the image which are then treated as probability distributions. A slightly different approach was taken in [161], whereby attraction is guided by the saliency distribution of the visual scene and the salient features are predefined probabilistically by a set of "interesting objects". The approach in [164] is conceptually similar to [156], however salience is defined as the Kullback-Leibler (KL) distance between a pixel's region (centre) Gaussian distribution to its surrounding regional (surround) distributions. The model in [162] uses eye movement data to treat saliency as a classification problem and in essence learn it. For a detailed review of computational models refer to [145].

4.1.3 The IKN model

The IKN model [156] is a prominent bottom-up saliency algorithm with a biologically inspired architecture that originates from [165] and as described in

the previous section, from feature integration theory [155]. The ultimate aim in IKN is to construct a two-dimensional topographic saliency map. The saliency model returns salient object in four steps:

1. Extraction of features
2. Creation of feature maps
3. Combinations/normalisation of feature maps
4. Winner-Take-All (WTA) neural network

The following figure shows the model's structure:

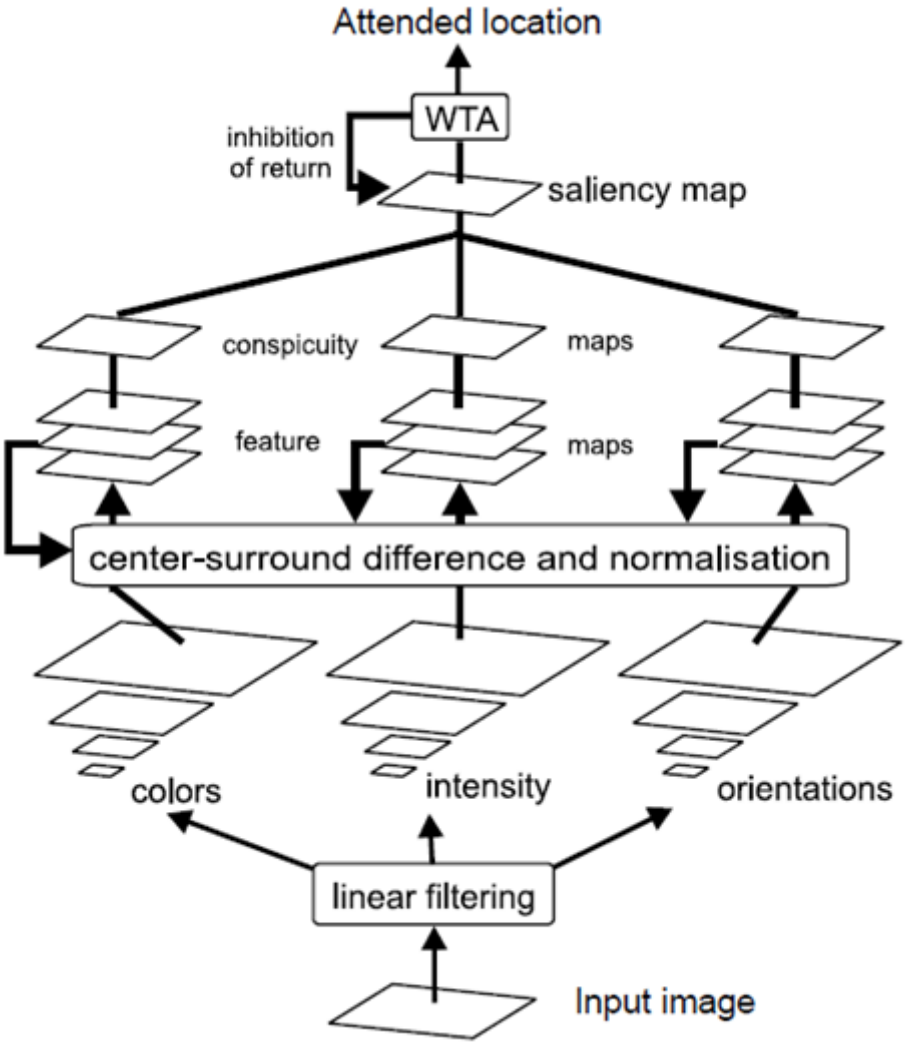


Figure 4-3: Layout of the IKN saliency model[156]**.**

The digital RGB input image is decomposed into three fundamental features of orientation, intensity and colour. These were chosen by the authors to be the foundational features of saliency maps in the primary visual cortex while the exact weight relationship between them has not been clarified. In biological

terms, numerous studies show that colour is a parallel evolutionary feature whose weight in human visual perception is subjective, given that cone wavelength sensitivity varies from person to person [166]. V1 shares strong feedback connections to higher cortical areas and in this case, through V2 and V3 to V4 which is the main colour processing centre. It has been proven that damage or deficiencies at the V4 part of the brain causes loss of colour perception and a condition called achromatopsia while vice versa, deficiencies in the V1 and V2 parts of the brain cause chromatopsia (i.e. the perception of colour without shape or form) [167]. V4 is categorised as part of the ventral processing stream and shares strong neuronal connections with the LGN and the pulvinar nucleus (often abbreviated as PUL, thought to be a router of various environmental information around the brain). This makes V4 more closely related to the top-down object-oriented tasks of the ventral stream rather than the scene-oriented mechanisms of the dorsal stream. However, whether early human visual perception is attracted more from colours or shapes or even intensity, is subject to a particular person's perception (i.e. whether colour acuity exceeds shape acuity or vice versa, etc), situation and task. It is perhaps, a fair consideration to keep an equal weight amongst these competing features for a salience model which simulates the first few milliseconds of topographic activity in the brain and decisions of preference do not yet affect the detection process.

The extractions of the early features from the input static colour image (or individual frames from a video) are achieved through the use of Gaussian pyramids [168]. Gaussian pyramids are a simplification of the gradual minimisation of the receptive fields of cortical cells as visual stimuli hierarchically progress in the brain. Gaussian pyramids are low-pass filters that progressively subsample the input image into nine spatial scales (0-8) altogether [156], which yield horizontal and vertical scale reductions ranging from 1:1 (0 scale) to 1:256 (8 scale) in octaves (consecutive powers of two i.e. 1, 2, 4, 8, 16, 32, 64, 128 and 256) and this is illustrated in Figure 4-4 and Figure 4-5. It is common practice in image processing to use these specific spatial scales within an octave Gaussian pyramid framework however a link with biology cannot be substantiated. The main reason is that the exact relationship between the receptive fields of simple and complex cells in all different layers and cortical areas in the brain, is unknown and if it exists there is strong evidence that it is in fact nonlinear [116].

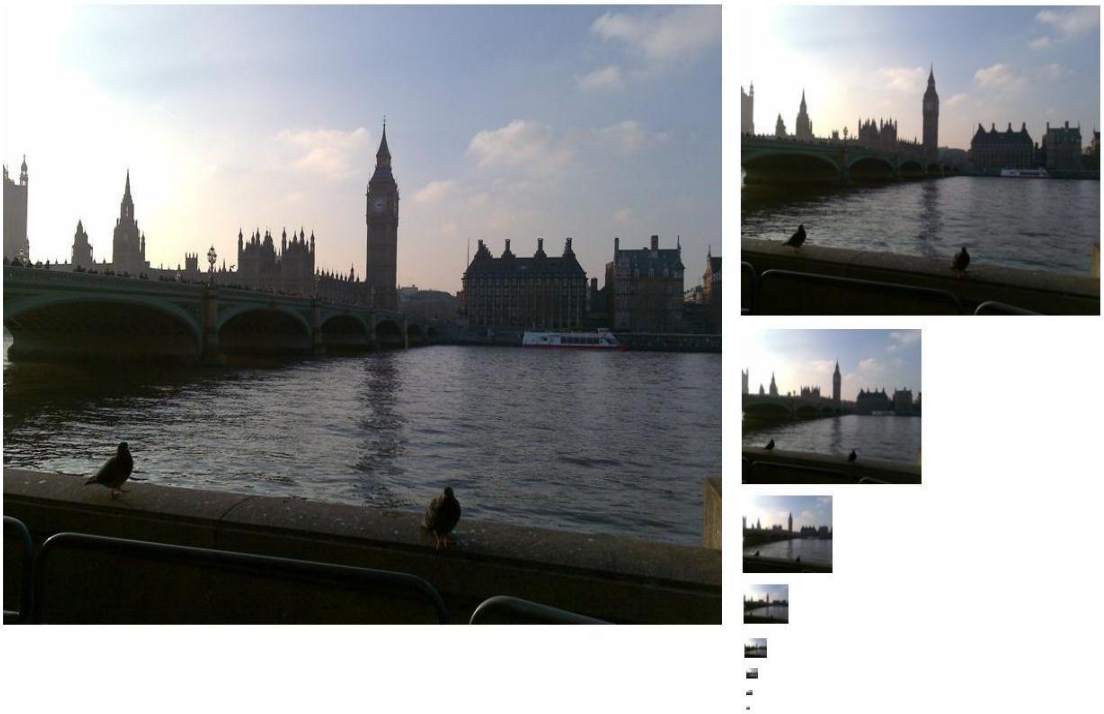


Figure 4-4: An illustration of an input image (left image, originally 800x600 pixels) gradually scaled down in 8 spatial scales (right) using a Gaussian pyramid.

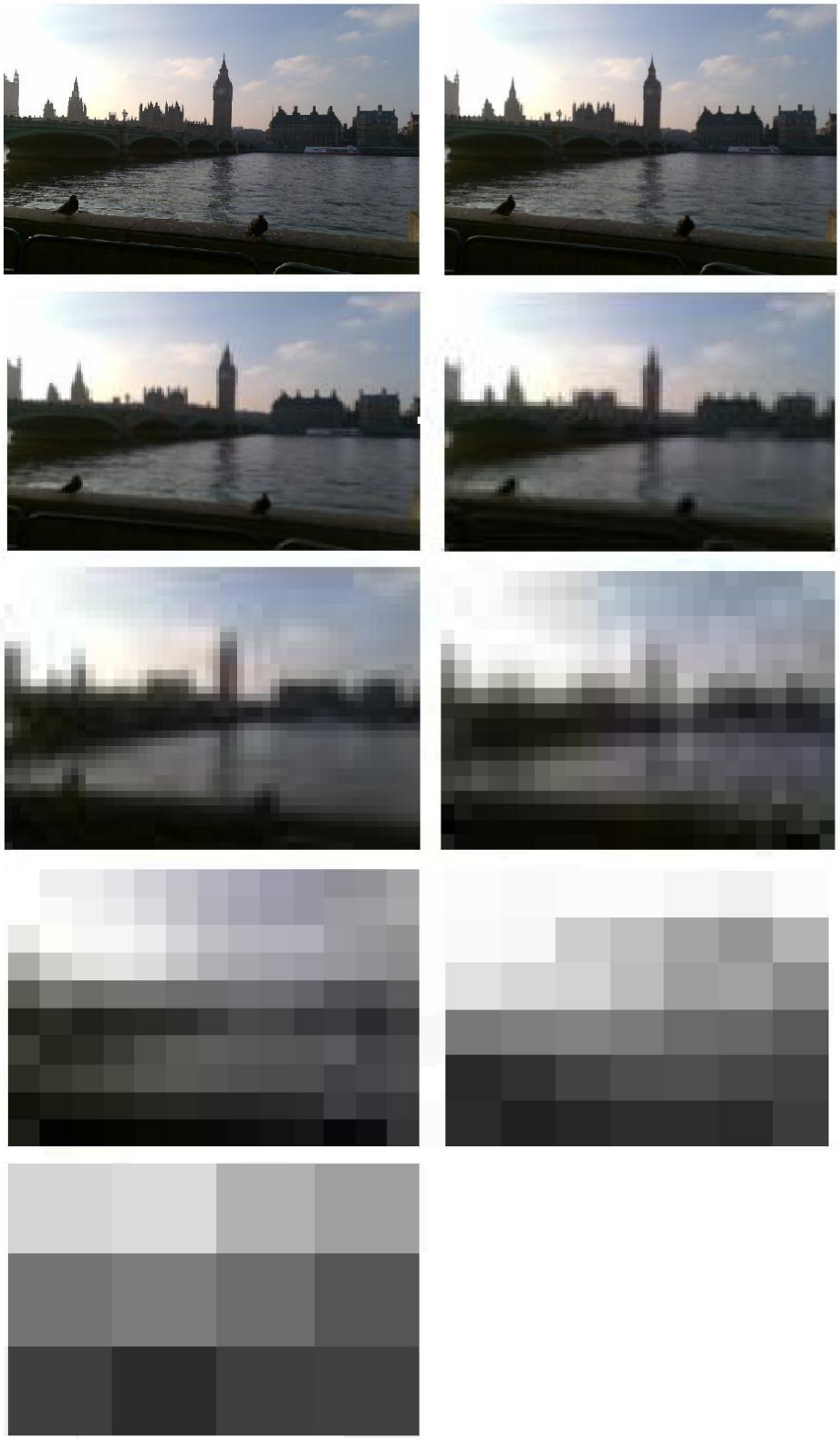


Figure 4-5: The pyramid of figure Figure 4-4 portrayed here in the same scale. From left to right, at the top row the input image is progressively scaled down according through a Gaussian pyramid in 8 spatial scales.

The first feature that is extracted from the input image in [156] is achromatic intensity (*I*). There are various ways to extract intensity depending on the colour model of the input image or the desired colour model (e.g. decolourisation methods [169] but in this biological-like approach intensity in a RGB input image, is given by [170]:

$$I = \frac{(r + g + b)}{3} \qquad \textbf{(4-1)}$$

From equation (4-1), a Gaussian pyramid for the intensity *I(σ)* is computed, where *σ = [0…8]*. Subsequently, the *r, g* and *b* channels of the input RGB image (or individual frame of a video) are normalised by *I* in order to remove the intensity from the colour information. This normalisation which resembles lateral inhibition is only applied at locations where the intensity is at least 1/10 of the maximum intensity (I_{max}) value of the whole image (equations (4-2), (4-3), (4-4)).

$$r = \begin{cases} \frac{R}{I} & for\ I > {I_{\max}}/{10} \\ 0 & otherwise \end{cases} \qquad \textbf{(4-2)}$$

$$g = \begin{cases} \frac{G}{I} & for\ I > {I_{\max}}/{10} \\ 0 & otherwise \end{cases} \qquad \textbf{(4-3)}$$

$$b = \begin{cases} \frac{B}{I} & for\ I > {I_{\max}}/{10} \\ 0 & otherwise \end{cases} \qquad \textbf{(4-4)}$$

The value of 10% was chosen as the assumption is that hue variations of low luminance do not result in salient features [156]. Therefore, four Gaussian pyramids *R(σ), G(σ), B(σ)* and *Y(σ)* are further created for each of the four broadly tuned colour channels (for the yellow channel negative values are set to zero) using the equations below [156]:

$$R = r - \frac{(g + b)}{2} \qquad \textbf{(4-5)}$$

$$G = g - \frac{(r + b)}{2} \qquad \textbf{(4-6)}$$

$$B = b - \frac{(r + g)}{2} \qquad \textbf{(4-7)}$$

$$Y = r + g - 2(|r - g| + b) \quad \textbf{(4-8)}$$

Lastly, the local orientations are computed from the first intensity map (from the input spatial scale 0) and four Gabor pyramids are extracted centred on each of the orientations examined, *θ = 0, 45, 90, 135* degrees. This extraction procedure relates to orientation selectivity that simple and complex cells exhibit in the primary visual cortex. In reality, tuning to only four orientations compared to biological cognition is a crude approximation.

Following the extraction of pyramids for the three features, different scales are chosen within those pyramids for the centre-surround computations. More specifically, the IKN creators, follow the "centre-surround" notion described previously, where the centre of a receptive field is defined as a pixel at scales *c = (2, 3, 4)* and the surround of a receptive field is defined as the pixel in coarser scales *s = c + d*, where *d = (3, 4)* [156]. The centre attention at spatial scales 2, 3 and 4 are an estimate of the visual accuracy that simple and complex cells show in V1. From the fourth scale onwards, the image's aspect ratio has been reduced to a point where the spatial scales simulate receptive fields at higher cortical areas. The surround is expressed in terms of higher spatial scales where visual detail has been removed (although not lost) and provides global information from the neighbourhood or the area around the centre.

Multi-scale feature extraction is achieved by interpolating the coarser scale to the finer and point-by-point subtraction. Hence, the intensity feature maps *I*, are obtained by:

$$I(c,s) = |I(c) \Theta I(s)| \quad \textbf{(4-9)}$$

In equation (4-9), *Θ* symbolises the centre-surround across-scale differences. The double colour opponent system between the colour pairs red/green and blue/yellow is used by the IKN model to create another set of maps for the colour feature extraction purposes. The centre-surround equations for these pairs are [156]:

$$RG(c,s) = |(R(c) - G(c) \Theta G(s) - R(s))| \quad \textbf{(4-10)}$$

$$BY(c,s) = |(B(c) - Y(c)) \Theta (B(s) - Y(s))| \quad \textbf{(4-11)}$$

To extract the orientation centre-surround feature maps *O* at *θ = (0^o, 45^o, 90^o, 135^o)*, are obtained [156]:

$$O(c,s,\theta) = \left|O(c,\theta)\Theta O(s,\theta)\right| \qquad \textbf{(4-12)}$$

Equation (4-9) creates six features maps for intensity, equations (4-10) and (4-11) create in total 12 feature maps, 6 colour feature maps each and equation (4-12), constructs 24 orientation feature maps, 6 for each of the orientations. So, the total 42 feature maps for all three features can now be combined to form the three conspicuity maps which lead to a combined conspicuity map. In order to calculate the conspicuity maps and since the IKN model lacks a top-down mechanism to decide where to direct attention, a normalising operator *N* is introduced. This operator promotes and highlights the areas where attention is most likely to be drawn which in effect scales and prepares the extracted features for further processing in unison. The normalising operator acts in three steps [156]:

1. Find the location of each map's global maximum *M* while calculating the average $\overline{m}$ for the remaining local maxima.
2. Normalise all values in each of the maps using the range *[0...M]* to eradicate amplitude differences.
3. Globally multiply the maps by $(M-\overline{m})^2$.

If amplitude differences between values are substantial then activation peaks emerge that promote the salient objects clearly across all feature maps and consequently, the areas of interest can be identified. The use of the *N* operator is therefore imperative since it unites the different early features while closely simulates the V2 operation of the brain where features are added across the receptive fields of cortical cells before they emerge for further analysis. As one would expect even with humans, if a salient object is not present in a relatively uniform scene the maps produce several accumulations of features across which they do not promote a certain area of attention.

At scale $\sigma = 4$ (this boundary scale lies in between the finer and coarser scales), and using the normalisation operator explained above, all 42 feature maps are combined by an across-scale addition $\oplus$ and point-by-point addition using the equations below [156]:

$$\overline{I} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c=4} N(I(c,s)) \qquad \textbf{(4-13)}$$

$$\overline{C} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} [N(RG(c,s)) + N(BY(c,s))] \qquad \textbf{(4-14)}$$

$$\bar{O} = \sum_{\theta \in \{0^o, 45^o, 90^o, 135^o\}} N\left(\bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(O(c,s,\theta)) \right) \quad \textbf{(4-15)}$$

Finally, the three conspicuity maps of intensity colour and orientation are further normalised and summed into a single saliency map *(S)* [156]:

$$S = \frac{1}{3}\left(N(\bar{I}) + N(\bar{C}) + N(\bar{O})\right) \quad \textbf{(4-16)}$$

Having obtained the saliency map of the most active locations from equation (4-16), then the map, at scale four, is ready to be passed through a threshold Winner-Take-All neural network such that values x_i of the saliency outputs correspond to values y_i of a neural network with hypothetical neurons that fire as soon as the required synaptic input value has been met (x_j) [165]:

$$\mathrm{y_i} = 0 \,\mathrm{if}\ \mathrm{x_i} < \mathrm{max_j x_j} \quad \textbf{(4-17)}$$

$$\mathrm{y_i} = \mathrm{f}(x_i) \,\mathrm{if}\ \mathrm{x_i} = \mathrm{max_j x_j} \quad \textbf{(4-18)}$$

In equation (4-18), f is any increasing function of x_i. The rate of the increasing function value is measured such that the salience location which reaches it first becomes the first region of interest followed by the locations that reach the threshold value. This maximum threshold value is set to a predetermined value (typically 10-20%) below the global maxima. This operation in[165], is also seen as a charging capacitor's function. Finally local areas are being inhibited if already activated, an action which avoids duplication of the salient location[156]. An example illustration, of the IKN algorithm as created using MATLAB for a single frame from a road scene video is shown below in Figure 4-6:

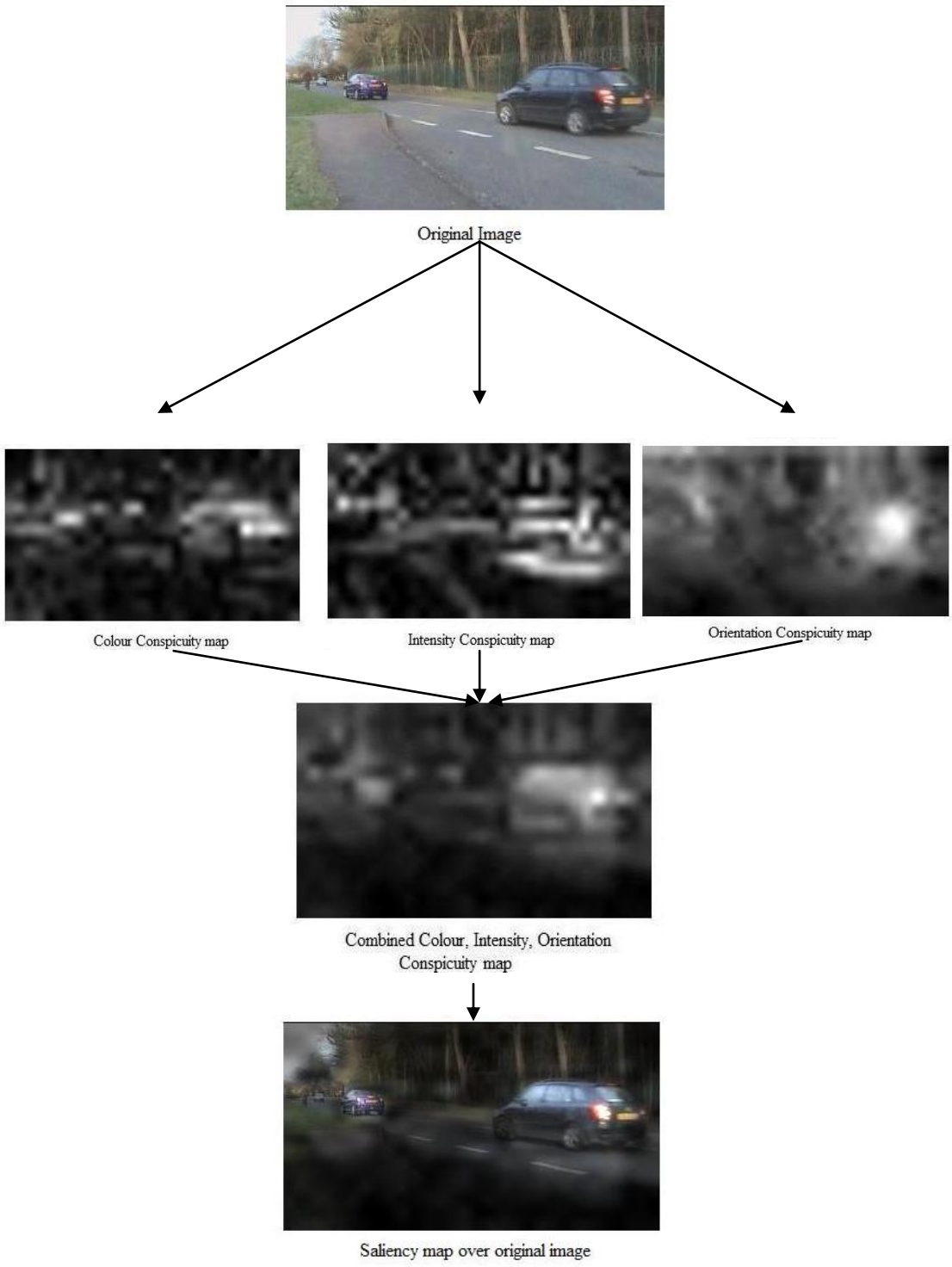


Figure 4-6: An example of the Itti/Koch/Niebur algorithm using a video's frame as an input image. The last image depicts the saliency map overlayed on the original image, shadowing any "uninteresting" areas of the image

4.1.4 Graph- Based Visual Saliency

The Graph Based Visual Saliency (GBVS) model [163] (or Graph-Based Saliency map, GBSM) is a more recent bottom-up saliency model that shares many similarities with the previously examined IKN model structure. GBVS has acquired its name for its use of directed graphs. Graphs here refer to collections of objects also known as “vertices” interconnected via “edges” [171]. In essence, these graphs define distance relationships between nodes i.e. distance between saliency regions. These distance maps substitute the pixel size feature maps in the IKN approach, in an attempt to unify smaller saliency regions together over the subsequent operations. More specifically, its feature extraction step (*s1*) follows the same principles and procedures as the IKN model. A different neurological approach (i.e. a graph is viewed as a collection of neurons instead of nodes) is taken for the activation step (*s2*) and normalisation/combination (*s3*) stages.

As in the IKN model, the first task is to create Gaussian pyramids for the extraction of the early features. The number as well as the spatial scales themselves for the Gaussian pyramids of colour, intensity and orientation in GBVS have the exact same definition as in the IKN model but in addition to these, GBVS offers new optional features of contrast, flicker and motion. Contrast in GBVS, following the notions of [172], is defined as the variance, i.e. the expected square deviation of a pixel’s value from its mean, of pixel intensities (normalised in the range *[0, 1]*) across an input image. In comparison with the intensity feature where colour is decoupled from intensity values, contrast takes into account the hue variations across the image. So, the across-scale differences *c = (2, 3, 4), s= c +d, d = (3, 4)* are found by:

$$R(c,s) = \left|R(c)\Theta R(s)\right| \qquad \textbf{(4-19)}$$

Flicker in GBVS is used for consecutive frames of video sequences and is defined as the absolute difference of values between two images. By convention, it may be also used to find the difference between images since it is a subtraction feature:

$$F_n(c,s) = \left|F_1(c,s)\Theta F_2(c,s)\right| \qquad \textbf{(4-20)}$$

Lastly, the motion feature, like flicker is only used for frames of video sequences and uses Gabor filters between two consecutive frames to find the direction of shift in pixel values according to angles of *0, 45, 90* and *135* degrees.

$$M_n(c,s,\theta) = \left|M_1(c,s,\theta)\Theta \mathrm{M}_2(c,s,\theta)\right| \qquad \textbf{(4-21)}$$

The use of contrast accounts for 6 extra feature maps, flicker for another 6 and motion for an extra 24 feature maps.

For the activation step (s*2*), assume any of the extracted feature maps as $M:[m\times n]\to\mathbb{R}$, where in the *M* array *[m] = {1, 2…m}* and *[n] = {1, 2…n}*. Initially, the model forms a directed cyclic graph across all its locations in [m x n] which is going to be a lattice and if all of the feature maps are examined collectively then it is a hierarchy (multi-resolution) of lattices. After the nodes have been created, it is important to obtain the weight of the outbound edges of these nodes.

Assume two nodes, *M(i, j)* and *M(p, q)* of a lattice derived from a feature map. It is then required to compute an activation map $A:[m\times n]\to\mathbb{R}$ in which the dissimilarity of values of *M (i, j)* with respect to *M(p,q)* would respond to values of *A*. In GBVS, the use of a geometric dissimilarity metric (d) between two different locations *M(i, j)* and *M(p, q),* is given as a choice between three optional equations[163]:

$$d\big((i,j)\,\|\,(p,q)\big)=\left|\log\frac{M(i,j)}{M(p,q)}\right| \qquad \textbf{(4-22)}$$

$$d\big((i,j)\,\|\,(p,q)\big)=M(p,q) \qquad \textbf{(4-23)}$$

$$d\big((i,j)\,\|\,(p,q)\big)=\left|M(i,j)-M(p,q)\right| \qquad \textbf{(4-24)}$$

Equation (4-22) simply means the ratio between two quantities measured on a logarithmic scale. Equation (4-23) transfers the value of *M(i, j)* to *M(p, q)* while accumulating it (mass concentration) and equation (4-24) is the difference, or more specifically the salient difference, between two nodes. Clearly, high values of these equations correspond to high values in *A*. In this fully–connected directed graph G_A that has been considered so far, every node of the array *M*, *(i,j)* $\in$*[m x n]* is connected with the rest of the *(n-1)* nodes and assuming an assigned weight *(w)* of a directed edge from node *(i,j)* to node *(p,q)*[163]:

$$w_1\big((i,j),(p,q)\big)=d\big((i,j)\,\|\,(p,q)\big)\cdot F(i-p,j-q) \qquad \textbf{(4-25)}$$

In equation (4-25), *F (α, b)* = $exp\left(-\frac{a^2+b^2}{2\sigma^2}\right)$, where *σ* is the free parameter (given as a fraction, 1/10 to 1/5, of the feature map width), represents the distance matrix between the nodes. Therefore, the weight as given in equation (4-25), implies that high values for an activation map *A* in two nodes *M(i, j)* and *M(p, q)*, are proportional to the distance between them. Intuitively, if there is

similarity between a node and its surrounding nodes then values in *A* do not vary as much.

A Markov chain on G_A can be created by normalising the weights of all outbound edges of each node to unity and drawing the equivalence between nodes and states, edges weights and transition probabilities [163]. In theory, if the resultant Markov matrix was to be repeatedly multiplied by a uniform vector then this would yield the principal eigenvector of the matrix while some equilibrium distribution (i.e. a state of convergence where all mass has been accumulated) will be met after *K* ($K<<n^2$, *n* being the number of nodes) iterations. However by default for the model, in this activation step *(s2)* the calculations are done only once. Finally, at this stage the number of feature to activation maps has not changed so for example for features of colour, intensity and orientation there are equally 42 activation maps.

In [163], the next step *(s3)* is named "Normalising an activation map". This is quite misleading as it simply, iterates one more time (or as many iterations someone so chooses) the above process of step *(s2)*. The objective behind the separation of the same process into two stages is to emphasise the importance of concentrating mass (weights) on activation maps even further because if activation maps are combined without further mass concentration, the resulting saliency map appears too uniform.

So, assuming the previous activation map $A:[m\times n]\to\mathbb{R}$ as the input then similarly to G_A, G_N graph can be constructed such that weights are:

$$w_2\big((i,j),(p,q)\big)=d\big((i,j)\,\|\,(p,q)\big)\cdot F(i-p,j-q) \quad \textbf{(4-26)}$$

Naturally, the choice of a dissimilarity metric still exists and again a Markov chain is present by normalising the weights of the outbound edges to 1. For features of colour, intensity and orientation as previously mentioned the number of activation maps remains unchanged to 42 in total. It should be noticed that the normalisation step that was just introduced, has completely substituted the *N* operator of the IKN algorithm. Simply, if further attention concentration is required, one has to increase the number of iterations.

Subsequently, the normalised activation maps are across-scale added to form a master saliency map in a way similar to the IKN model (equations for colour, intensity and orientation are identical to the IKN) while the extra 3 features of contrast *(R),* flicker *(F)* and motion *(M)* are across scale added as follows:

$$\overline{R} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c=4} R(c,s) \qquad \textbf{(4-27)}$$

$$\overline{F} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c=4} F(c,s) \qquad \textbf{(4-28)}$$

$$\overline{\overline{M}} = \sum_{\theta \in \{0^o, 45^o, 90^o, 135^o\}} \left(\bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} (M(c,s,\theta)) \right) \qquad \textbf{(4-29)}$$

The saliency map is simply as shown previously (if all features were to be chosen together):

$$S = \frac{1}{6}\left((\overline{I}) + (\overline{\overline{C}}) + (\overline{\overline{O}}) + (\overline{R}) + (\overline{F}) + (\overline{\overline{M}}) \right) \qquad \textbf{(4-30)}$$

After obtaining the saliency map, GBVS does not include any further steps. As seen in the previous section, the IKN algorithm has a Winner-Take-All neural network to compensate for the lack of a priority mechanism that shows where attention is attracted sequentially, much like human eye movements prove. However, this mechanism could also be implemented in GBVS to sequentially show where mass concentration on the lattices of the directed graphs is accumulated.

Markov chains and in consequence Bayesian networks, have been known to closely simulate neuronal communications [173], [174] and there is growing use of Bayesian theory in neuroscience and cognition. In theory GBVS is more biologically plausible than IKN with its use of lattices, similar to layers of neurons. The following figure (Figure 4-7) show examples of features in GBVS. The motion and flicker features are presented in great detail in section 5.1.

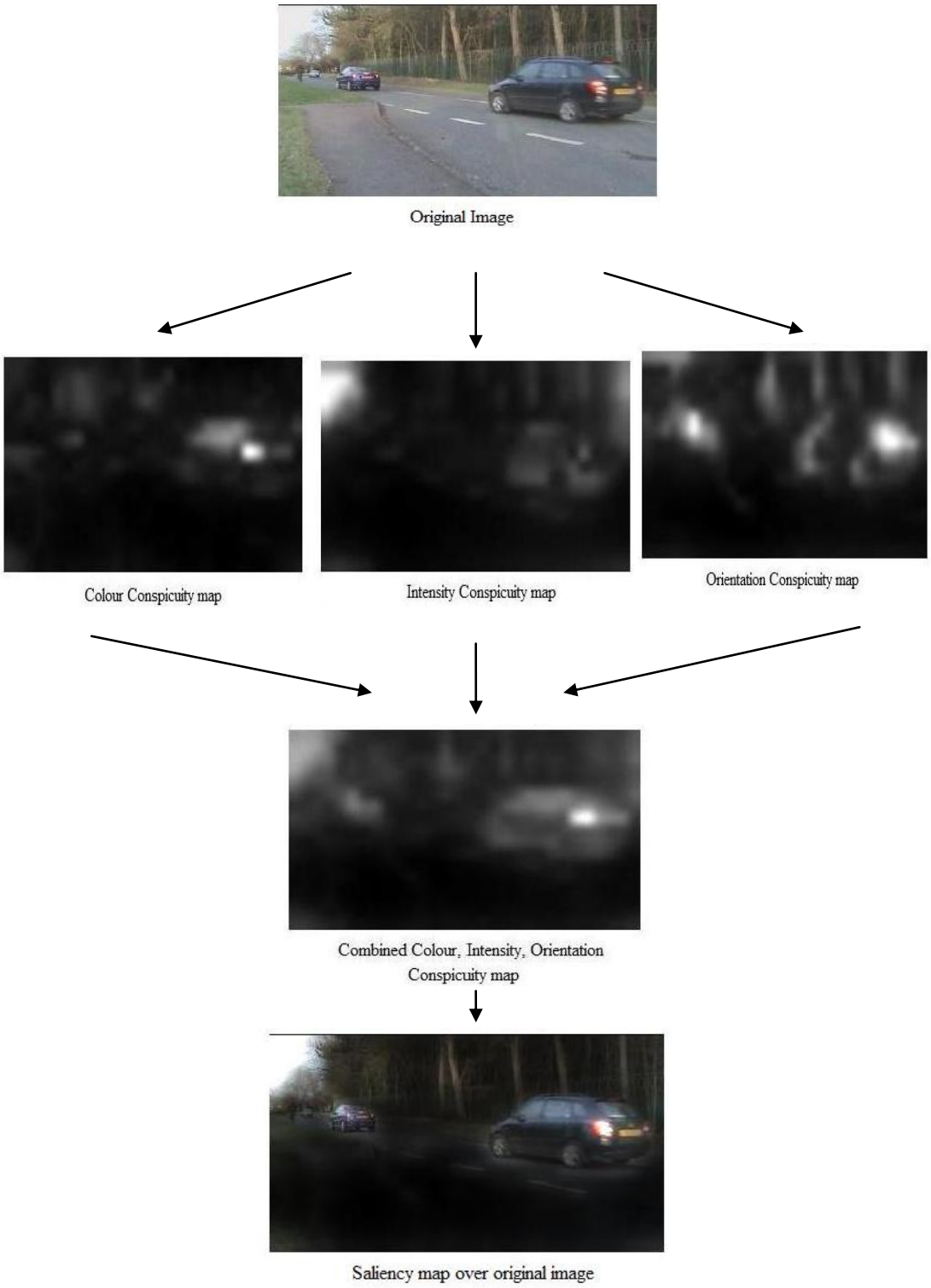


Figure 4-7: An example of the GBVS algorithm using a video's frame as an input image using 3 features (colour, intensity, orientation). The last image depicts the saliency map overlayed on the original image, shadowing any "uninteresting" areas of the image.

4.2 Ventral Stream and Object recognition

4.2.1 Biological overview

Visual data that departs the ventral V3 area, enters the ventral stream or “what” pathway and first reaches area V4. More than half of the number of neurons in V4 exhibit sensitivity to colour while the remaining neurons deal with spatial frequency and orientations. This suggests that V4 is an early centre for processing the association of form and size to colour [124]. Even though neurons at this point contain large receptive fields, they are object-driven (section 4.1.2.2) i.e. OBA, rather than scene-driven like neurons in the V3 dorsal and V7 areas. Area V8 is adjacent to V4 and it is exclusively colour selective [175], [176], making it a centre for colour perception and processing.

Neuronal responses from areas V4 and V8 are channelled through neurons with large receptive fields (at least four times larger than cells in V1 [177], possibly containing information of whole objects) to the temporal lobe (Brodmann areas 20, 21, 22 and 38) and more specifically the inferotemporal lobe (IT).

IT neurons have broad receptive fields, much broader than neurons in V1. Moreover, they show properties of scale and position invariance [120], [178], cue invariance [179] and rotation invariance [120]. The inferotemporal lobe is subdivided in the posterior inferotemporal (PIT), the central inferotemporal (CIT) and the anterior inferotemporal (AIT). PIT and CIT synthesize complicated objects (i.e. faces, letters etc) for detailed and complex representations [180] while AIT contributes in classifying and recognising these objects [181]. It has been shown that neurons in AIT retain object selectivity like early ventral stream neurons in V4 [182], [183].

Finally, long-term memory including object shapes (e.g. faces) as well as spatial memory and behaviour, is stored in the medial temporal lobes of the IT. Similarly to the parietal lobe, the temporal lobe maintains neuronal connections and feedbacks with the entire brain.

4.2.2 Object recognition models

Hierarchical architectures such as constellation models, multilayered convolution networks and generic object recognition, have become increasingly popular over the last years [184]. Hierarchical based generic object recognition studies have started to appear in recent years as a way to simulate the ventral pathway of the visual cortex. There had been attempts e.g. [185], [186] that did not expand much in large scale image databases but nevertheless had hierarchical structure exhibiting invariance to certain properties. Similarly, for

character recognition, convolutional networks employ principles of feedforward hierarchical structures of alternating layers[187].

A biologically plausible approach was achieved in HMAX (Hierarchical Model and X) [117] and its refinements in [184], [188]. This model's approach resembles the way that simple and complex cells in the visual cortex cooperate to achieve recognition. Object recognition follows a feedforward bottom-up structure via spatial pyramids (or hyperfeature stacks) [168] to create a feature dictionary of orientations that is used to achieve generic object recognitions. To date, apart from minor improvements [189] on this model or perhaps references to some aspects of biological vision, no other model claims a strictly biological object recognition architecture such as HMAX. It is because of its biological plausibility, computational simplicity and efficiency that HMAX and FHLib have become the foundation for object recognition in this work.

4.2.3 Hierarchical Model and X

Hierarchical Model And X (HMAX) is a bottom-up feedforward object recognition model that has been created to simulate the ventral visual processing stream. Unlike, any other model which has some biologically inspired portions incorporated in its architecture, HMAX claims a biologically exclusive structure and follows ideas first presented in [117]. The basic idea of this hierarchical model is to produce a position and scale invariant representation of objects which can be learned, and subsequently used as a visual vocabulary for recognition and tracking much similar to the visual cortex of primates.

According to [184], HMAX summarises the following properties in neuroscience:

1. Ventral visual processing is hierarchical, progressively increasing its invariance to object position and scale.

2. As the layer levels along the hierarchy increase so do the neuronal receptive fields.

3. The initial processes (i.e. the first microseconds) of visual information in the ventral pathway, are feedforward (i.e. bottom-up).

4. Learning occurs at the top cortical layers i.e. the inferotemporal (IT) and prefrontal lobe (PFC).

Properties 1 and 2 arrive from the implementation of the general theory behind cortical cells presented in section 3.4. Generally, simple cells with strong orientation sensitivity and small receptive fields progressively combine with

complex cells which therefore have larger receptive fields and are not orientation sensitive. This general principle is continued until enough information has accumulated to create even more refined spatial features such as edges, corners, shapes, sizes, foreground, background, distance and depth associations of objects in the scene. It is known [190] that between 40ms-100ms and after the visual information has reached the inferotemporal cortex, that learning and recognition occurs in a feedforward manner and further tasks take place thereafter, as explained in properties 3 and 4.

Originally, in HMAX, nine simulated layers were formed to tackle with object recognition. These cortical-like layers alternated between simple and complex units essentially fulfilling similar operations in a progressive way. So the original sequence of layers (Figure 4-8) was formed by *S1, C1, S2, C2, S2b, S3, C2b, C3* and *S4*. However, according to [190], the necessity of all these layers was not completely justified and the model could be reduced to a looser hierarchy of 4 layers (*S1, C1, S2, C2*) described in detail in this section.

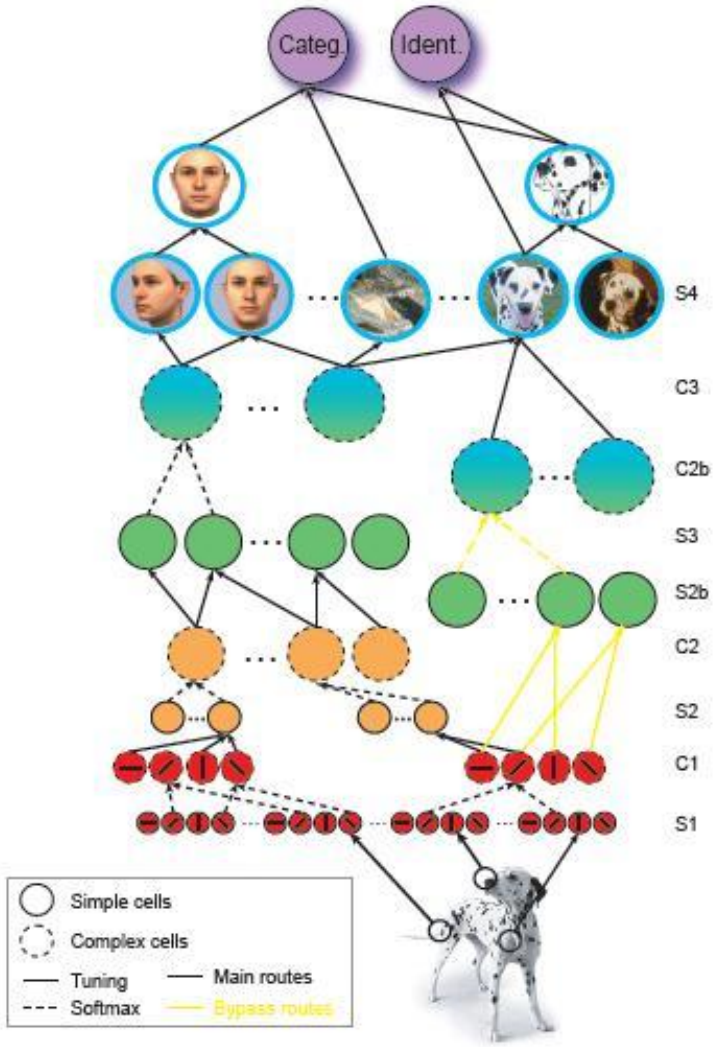


Figure 4-8: The initial conception of the HMAX architecture. Simple and complex layers of units alternate via tuning and pooling operations to provide further invariance to the spatial information of an object [190]**.**

In HMAX, the only feature used for recognition is the spatial orientation data of an object. Simple units *(S1)*, from the homonymous simple cells (sections 3.4.1, 3.4.2), perform the first linear tuning operations in a Gaussian-like way, to bars of a specific orientation. In engineering terms this is realised as a Gabor filter, which is simply a harmonic function multiplied by a Gaussian envelope, returning the edges of shapes as an edge detector would. *S1* layer units are fed with an input image which has to be in gray scale and image height always scaled to 140 (or the model converts a colour image to gray, since the algorithm does not account for colour or any other top-down feature) and Gabor filters are applied as given from equations (analysed further in 3.4.2) below:

$$F(x,y) = \exp\left(-\frac{x_o^{\,2} + \gamma^2 y_o^{\,2}}{2\sigma^2}\right)\left(\cos\frac{2\pi}{\lambda}x_o\right) \quad \textbf{(4-31)}$$

$$x_o = x\cos\theta + y\sin\theta \quad \textbf{(4-32)}$$

$$y_o = -x\sin\theta + y\cos\theta \quad \textbf{(4-33)}$$

The variables (x_o, y_o) in equation (4-31), refer to a *S1* receptive field in a 2D coordinate system. The aspect ratio is at $\gamma = 0.3$ and the orientations θ at 0^o, 45^o, 90^o and 135^o. On the other hand, effective width σ and wavelength λ, are varied in a particular way (explained below) to match the parafoveal simple cells of V1 operations as presented in section 3.4. The created *S1* Gabor filters have 16 different sizes (chosen by the authors) that range from 7x7 to 37x37 pixels in steps of two pixels (i.e.7x7, 9x9, 11x11...37x37) for each of the four orientations (16 scales x 4 orientations = 64 *S1* maps) forming a pyramid of filters. In this filter pyramid, varying σ and λ, ensures that the sparse representation of the Gabor filtered image is kept acute across all 16 scales and thus preserving the local maximisations for the next *C1* layer. However, a drawback to this approach is that as these parameters are varied, new local maxima are being created. It is argued in [184], that the Gaussian pyramid approach although rightfully considered for fine to coarse representations (as used in IKN and GBVS), would not preserve the sparse inputs of a Gabor filtered image without dilutions (i.e. losing the local maxima). Parameters σ and λ are varied by ad-hoc empirical relationships [184]:

$$\sigma = 0.0036 \cdot RFsize^2 + 0.35 \cdot RFsize + 0.18 \quad \textbf{(4-34)}$$

$$\lambda = \frac{\sigma}{0.8} \quad \textbf{(4-35)}$$

In equation (4-34), *RF* is the receptive field as given by the filter size (i.e. 7 for 7x7 filter size, 9 for 9x9 etc) (Appendix A).

The following *C1* layer corresponds to cortical complex cells which have a greater receptive field than the simple cells and pool *S1* units of the same orientation and scale band. Firstly, in HMAX the 16 *S1* filters are arranged in pairs to 8 bands of an individual orientation (shown in table of Appendix A). Each band is sub-sampled by taking the MAX operation over a grid cell with size N^{Σ} x N^{Σ} (shift tolerance) and the maximum over a particular band's members (scale tolerance) (where Σ[1] is the scale band and *N* symbolises the

[1] Note that Σ is a label.

neighbourhood) for each of the orientations separately. For example, for band 1 which contains two filter sizes (7x7, 9x9), a spatial maximum is computed over an 8x8 cell grid and across the two filters of the same orientation. The cell grid and the scale bands determine therefore, the range of spatial and size pooling of the *S1* inputs. The size of the cell grid is incremented by 2 pixels per band (i.e. 8x8 for 1st band, 10x10 for the 2nd etc) and choosing this size has not been clarified by [184] and therefore assumed to be chosen empirically.

The MAX operation (i.e. non-linear summation of responses) as illustrated in Figure 4-9, can be applied in simple biologically plausible models [184], [190], [191]. As explained in detail in section 3.4.3 of this thesis and through experiments on cortical complex cells [192], the MAX-operation is not a biologically universal relationship for complex cells. However, many cells do exhibit this MAX-like behaviour and it was found to be independent of the distance between bars of orientation and relative amplitude of cells' responses [192]. Even so, the importance of the MAX-operation in a generic object recognition system which has to be invariant of position and scale is great. The MAX relationship is given by:

$$r = \max_{j=1\ldots m} x_j \qquad \textbf{(4-36)}$$

In equation (4-36), *r* is the *C1* unit response (i.e. the maximum amplitude of the *S1* inputs) of its *(m)* S1 vectors $(x_1 \ldots x_m)$.

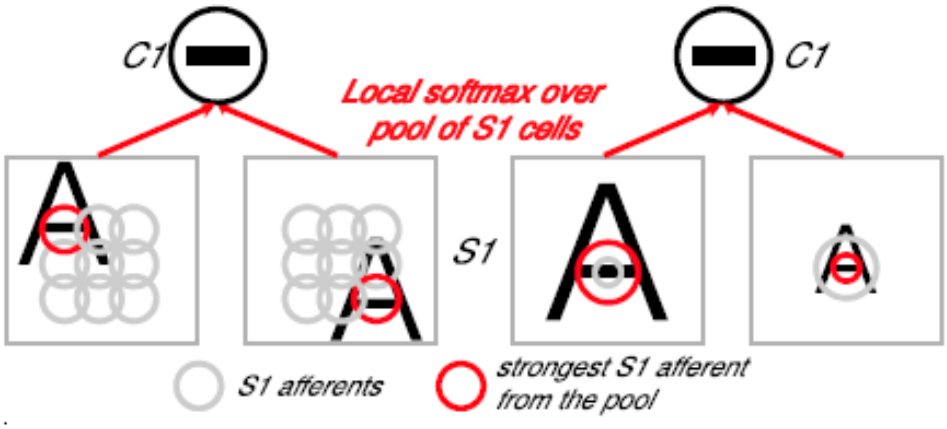


Figure 4-9: Example of a *C1* unit max operation. Left *C1* unit finds the strongest S1 vector invariant to shift, right *C1* unit finds the strongest *S1* vector between two different scales (softmax has the same definition as max)[190]**.**

In order to be computationally efficient for the next stage, *C1* responses are not calculated for every possible location and may only overlap by an amount *Δs* as

given in the table of parameters (Appendix A). Two practical examples over the Gabor filtered image is shown in grayscale in Figure 4-10.

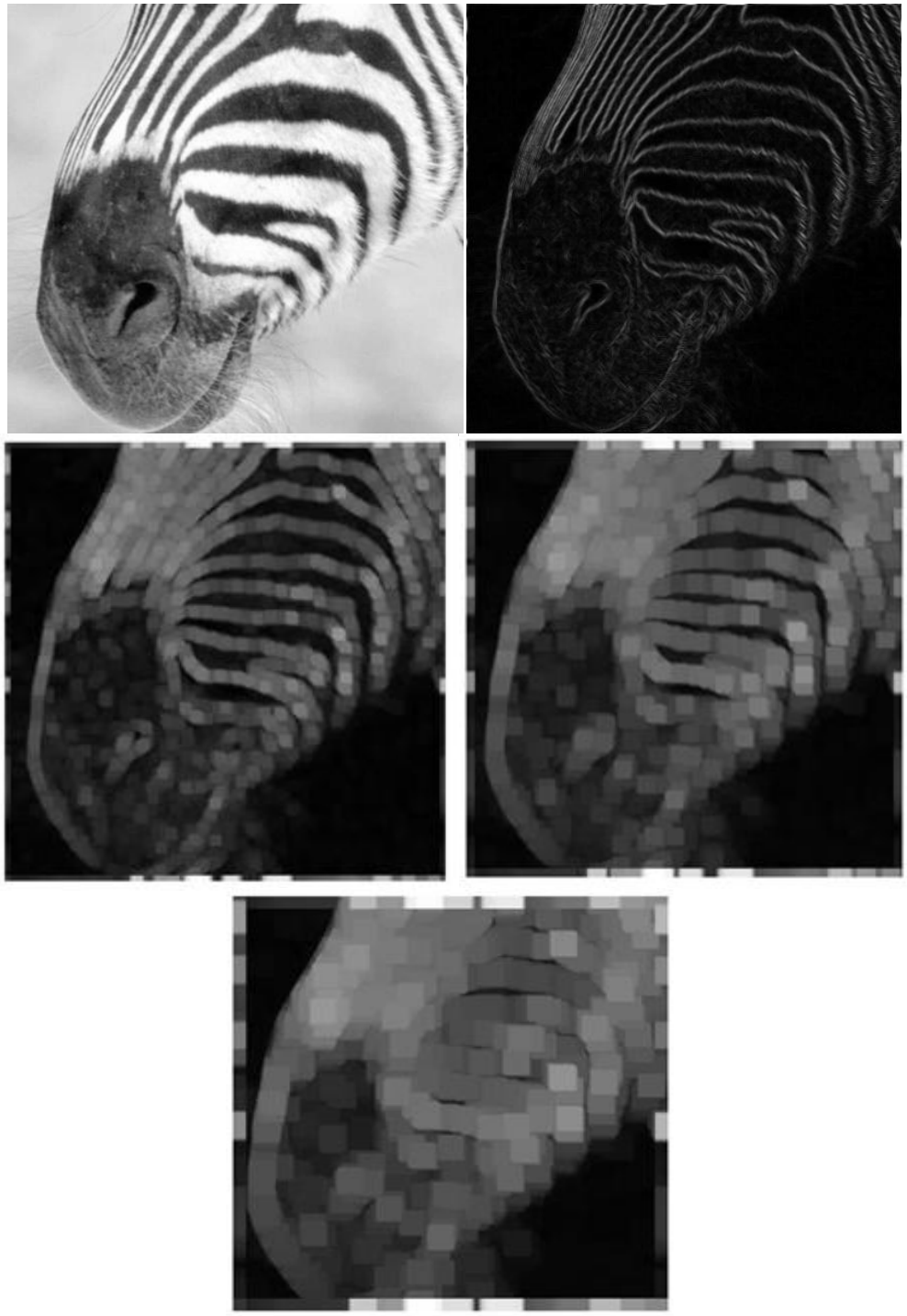


Figure 4-10: An example of a Max-operation applied over the Gabor-filter extracted image. Top left is the input image and top right is the first Gabor image obtained at four orientations. Middle left image is at band 1 (i.e. spatial pooling 8x8), middle right image at band 4 (14x14), bottom-left image at band 8 (22x22).

The *S2* layer, is hypothetically simulating the functions occurring as soon as object information has reached the inferotemporal (IT) area of the brain. During the training stage the user or operator of the algorithm may choose a number of *K* patches (also known as features) that are extracted at random positions from the *C1* maps of the training images so that $P_{i=1...K}$ have various sizes of $n_i \times n_i \times 4$ (i.e. where n_i can be 4, 8, 12 and 16, by default or any size, and 4 is the number of orientations). The input images should be fed from a positive dataset, i.e. images containing the object to be learned. At this stage, the *S2* units are simply passed on to the following *C2* layer. The extracted features are also called prototype patches or centres of the *S2* units and all together form a filter or universal dictionary or library of features that is used during the testing stage.

Alternatively, if the *S2* layer has been reached during recognition (after feature learning), then when a test image is applied, each of the previously stored S2 patches is convolved with the new $(C1)^{\Sigma}$ image, across all 8 scales and 4 orientations $(K \times 8(S2)^{\Sigma} \times 4)$ creating responses according to equation (4-37). Using a radial basis function, the Euclidean distance between the new input test patch (at the particular position and scale) and the stored prototype behaves in a Gaussian way. For a test image patch *X* of a scale *S*, the *S2* unit is found by [184]:

$$r = \exp(-\beta \| X - P_i \|)^2 \qquad \textbf{(4-37)}$$

In the equation above, *r* is the response of the *S2* unit, β is the sharpness of the tuning and P_i is one of the prototype features that have been learned during the training stage.

Finally, similarly to the *C1* layer, a new maximum operation is applied across all *S2* scale bands and orientations which make the *S2* units, shift and scale invariant *C2* global vectors. Naturally, the number of *C2* units during training depends on the number of chosen extracted features/patches. After feature learning, the *C2* vectors are fed to a linear classifier (e.g. SVM).

At the recognition stage, *S2* units represent the response or the stimuli of a test image against the stored training dataset. *C2* testing vectors are max-pooled vectors from the relationship $K \times 8(S2)^{\Sigma} \times 4$, i.e. each *C2* vector represents the maximum response to one of the stored prototypes. Intuitively, if a certain test image *C2* pattern closely matches a certain *C2* training category of the vocabulary then it most likely belongs to that category. Thus, the *C2* testing outputs are fed again to the same SVM and compared against the input training *C2* vectors. Figure 4-11, summarises the HMAX architecture explained above.

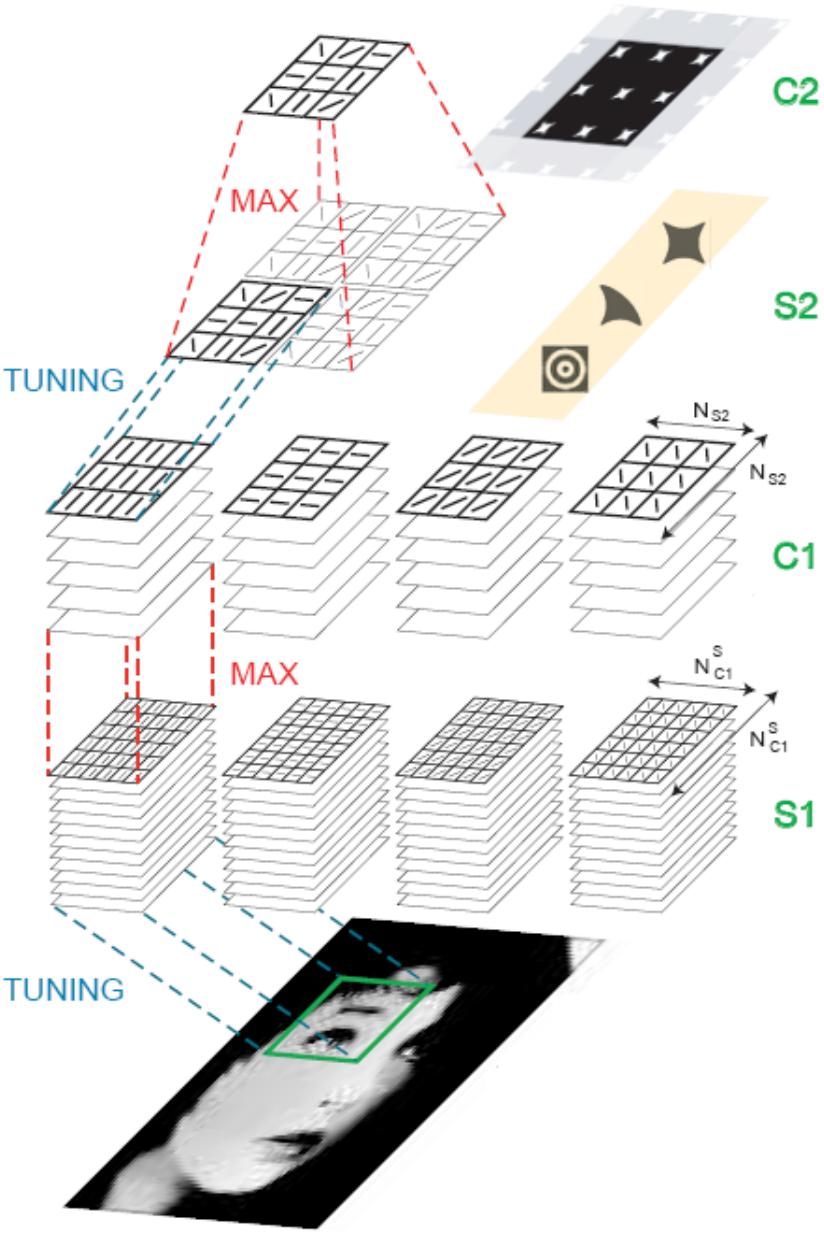


Figure 4-11: The HMAX architecture consisting of four layers (*S1, C1, S2, C2*). A position in the input gray image is Gabor-filtered *(S1)* in 16 scales of four orientations in each (image showing 8 for simplicity). *C1* units extracts local maxima over positions and scales while *S2* layer using an RBF function compares previously extracted patches using an RBF function. *C2* values are computed by taking a max over the *S2* results [184]**.**

4.2.4 Feature Hierarchy Library

Feature Hierarchy Library's (FHLib) architecture builds on HMAX, described in the previous section. However, there are differences between the two models and certain improvements are introduced with FHLib. FHLib is a bottom-up feedforward hierarchical model and like HMAX, consists of four layers *(S1, C1, S2, C2)* with only feature for recognition being again the spatial orientations of objects [189].

The input image is first converted into an intensity image (greyscale) using the equation below:

$$\left(I = {r + g + b}/{3} \right) \qquad \textbf{(4-38)}$$

Then while preserving its aspect ratio, the shorter side is scaled to 140 pixels. After preparing the input image, a pyramid with 10 spatial scales is created (including the input image), each being $2^{1/4}$ smaller than the last one, using bicubic interpolation. Bicubic interpolation takes the information of the original pixel and sixteen of the surrounding pixels (in the 4x4 neighbourhood) to determine the colour values of the new pixels created from the original. This concept is the first and main difference with HMAX which utilises a pyramid of Gabor filters with varying parameters but not a pyramid on the input image as FHLib.

The *S1* layer consists of Gabor filters of constant size 11x11 (filter size does not vary as in HMAX) which are placed at every position and at all scales on the input pyramid. Their construction follows equations (4-39), (4-40), (4-41) as below:

$$F(x,y) = \exp\left(-\frac{{x_o}^2 + \gamma^2 {y_o}^2}{2\sigma^2} \right)\left(\cos\frac{2\pi}{\lambda} x_o \right) \qquad \textbf{(4-39)}$$

$$x_o = x\cos\theta + y\sin\theta \qquad \textbf{(4-40)}$$

$$y_o = -x\sin\theta + y\cos\theta \qquad \textbf{(4-41)}$$

In equations (4-39), (4-40) and (4-41), θ is *0, 45, 90, 135* degrees and γ *=0.3,* σ*=4.5,* λ*=5.6* are kept constant, derived the empirical parameters of the HMAX table in Appendix A for band 2. The components of each of the Gabor filters are subsequently normalised to a mean 0 and a sum of squares to 1.

In the *C1* layer of FHLib, *S1* units are pooled to become shift and scale invariant *C1* units. For this reason, a max-operation takes place similar to HMAX. Each of

the four *S1* pyramids is convolved with a maximum cell grid of size 10x10. Therefore, the value of a *C1* unit is the maximum of a *S1* unit under the grid cell. The max filter is moved in steps of 5 pixels (1 in scale) giving a sampling overlap factor of 2. Since *S1* are in a pyramid structure the filter was kept at the same size for all scales.

$$r = \max_{j=1\ldots m} x_j \qquad \textbf{(4-42)}$$

The *S2* layer basic concept in FHLib is identical to HMAX although a pyramidal approach is used with a different response equation. Specifically, during training, the operator of the algorithm may choose a certain *K* number of prototypes *(P)* or features to be extracted from a positive training dataset. These prototypes are of progressive size 4x4, 8x8,12x12 or 16x16 for four orientations i.e. $n_i x n_i x 4$ and are randomly sampled from *C1* unit values that fall within that patch (for example, for the first patch size, 4x4x4 = 64 *C1* units). Like previously seen in HMAX, the user has no control over which features are extracted and their size. All features are stored in a common featurebook for the testing phase of the model.

During the testing stage, *S2* layer units behave almost like filters in which *C1* test input units are convolved to obtain their responses. An *S2* pyramid is generated with the same number of positions and scales as *C1* but at each position/scale exist *K* number of responses of that corresponding *C1* unit with respect to the prototype patches. These responses are subject to the following equation.

$$r = \exp\left(-\frac{\|X - P_i\|}{2\sigma^2 \alpha} \right)^2 \qquad \textbf{(4-43)}$$

In equation (4-43), *r* is the response, *X* is the input test *C1* patch, P_i is the *K* number of patches, σ is the standard deviation (set to 1, by default) and α is the normalising factor of the different patches. This normalization factor is responsible for reducing the higher dimensional space created by patches greater than 4x4 (i.e. 8, 12 and 16). The weight of the extra dimensions is reduced by $\alpha = (n/4)^2$, i.e. the ratio of the dimensions of *P* over the dimensions of the smallest patch [189].

Finally, the *C2* layer global units at the training stage are created by a max-operation bringing together *S2* units, under the form of *K* number of vectors across all positions and scales. For example, an *S2* patch size of 4x4x4 (which contains the 64 previous *C1* units) is joined across all orientations and all scales (5, since 10 *S1* original scales were reduced via *C1* by a factor of 2) according to the maximum responses into a single *C1* vector. The *K* number of *C2* vectors

is fed to a linear classifier, e.g. SVM. At the testing stage, the each of the K elements of the *C2* vectors represents the maximum response of the test image against the trained *K* prototype patches and can be used for classification in a SVM classifier so that its closest category can be found. Figure 4-12 summarises FHLib's architecture as explained above.

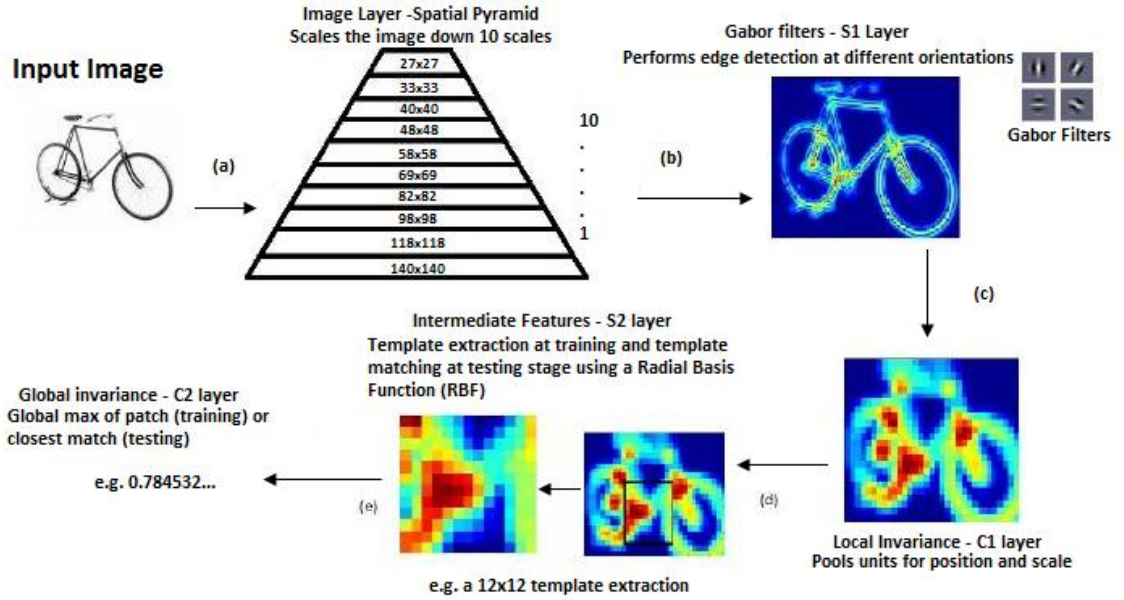


Figure 4-12: FHLIB's architecture. A pyramid of various resolutions of the image is followed by extraction of Gabor features in the S layers and max-pooling across the adjacent C layers. Subsequently spatial information is translated to feature vectors for classification.

To summarise the differences between FHLib and HMAX, in HMAX:

1. A pyramid approach is not used (different size filters are instead applied)
2. Parameters σ and λ, are varied from scale to scale at the *S1* layer
3. *S1* filters differ progressively
4. *C1* sub-sampling ranges do not overlap in scale
5. *S2* has no normalising α parameter

In FHLib, three further improvements were introduced. Inhibit *S1/C1* outputs, sparsify *S2* units and limit position/scale invariance in *C2* and these are described below.

Inhibiting *S1/C1* units (shown in Figure 4-13) is essentially an operation of suppressing their outputs so that dominant orientations compete at a particular location. For this reason an inhibiting parameter *h* is introduced which can be set between 0 and 1. The relationship between the minimum R_{min} and maximum responses R_{max}, over all orientations is given by (where *R* is the response of a unit at a particular location):

$$R < R_{\min} + h(R_{\max} - R_{\min}) \qquad \textbf{(4-44)}$$

As seen in equation (4-44), the inhibition level represents the fraction of the response range that gets suppressed. This antagonistic convention is in fact very similar to lateral inhibition (section 3.2).

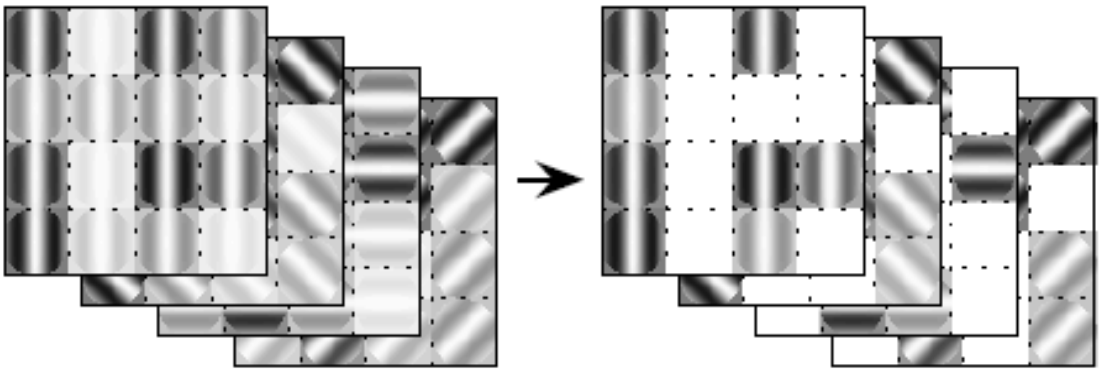


Figure 4-13: An example of inhibiting *S1/C1*, a 4x4 patch (single scale) at four orientations. The weaker responses (lighter) of the left set of original units are suppressed so that only the strong responses remain (darker) [189]**.**

Another enhancement introduced by FHLib is the sparsification of *S2* inputs, as depicted in Figure 4-14. As seen previously in this section *S2* units at the testing stage, *S2* units would obtain responses of *C1* units against the stored library of features. Sparsifying *S2* units is used under the notion of cortical cells being very selective of their inputs and thus instead of 4 orientations of *C1* units the input is actually one. To achieve this during the training stage, the identity and magnitude of the dominant orientation is stored at each of the n_i *x* n_i positions of a particular patch. For example, a 4x4 patch will correspond to 16 *C1* instead of 64. *S2* units become less sensitive to local clutter while the dimensionality of equation (4-44) is lower. At the same time, as proposed by [189] the number of orientations should be increased (in the FHLib from 4 to 12) to improve spatial accuracy.

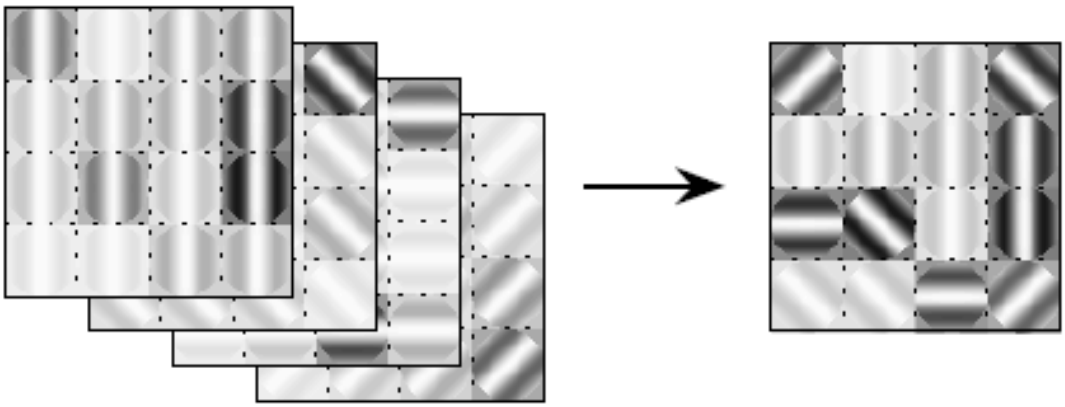


Figure 4-14: An example of sparsifying *S2* features. A 4x4 patch at four orientations, the left set of dense *S2* units of the base model shows the sensitivity to all orientations of *C1* units. The stronger responses on the right are sparsified (darker) and create a feature more sensitive to a particular orientation at each position [189]**.**

So far *C2* units in HMAX and FHLib base models have been treated as global position/scale invariant vectors which express the maximum response of a test image to all *S2* features. To address the issue of false positives or co-occurrence of features across different object categories some geometric information is passed to *C2* vectors. This brings the last enhancement of FHLib which is to limit the position/scale invariance of *C2* vectors. Under this improvement the locations of the sampled *S2* features from training images play a role with respect to the image size *($\pm t_p$%)* and scale *($\pm t_s$%).* Alternatively put, interesting features are assumed to be found around the centre of an image where visual attention is likely to be acute (section 4.1.1). In the next chapter this method of searching for *S2* features around the centre of an image is considered a drawback. Practically, objects of interest do not always appear in the centre of images or there may be more than one object at any time. This is remedied in the next chapter with the use of saliency maps. Some of the reasons for choosing saliency maps were: biological plausibility, ease of use, bottom architecture and compatibility, explained in more detail in the beginning of the next chapter.

5 MODELLING BIOLOGICAL VISION - SETUP

5.1 Overview

The work here expands primarily on the three computational models, IKN (4.1.3), GBVS (4.1.4) and FHLib (4.2.4). These original models were chosen as the foundation for visual attention (dorsal stream) and recognition (ventral stream), because of their:

1. **Biological plausibility.** In the chosen models, operations follow physiological and psychological discoveries closely. For example, edge detection for morphological representation is performed using the biologically plausible Gabor filters (3.4.2). Colour in the saliency models is introduced via colour opponent channels and centre-surround operations (3.2). Other features of motion, contrast and intensity (4.1.1, 4.1.3) are also inspired from biological mechanisms. The progressive hierarchical structure which characterises these models is an important aspect of information processing in mammalian brains (2.1, 3.4). In addition, FHLib uses maximum-pooling functions (3.4.3) to provide scale and position invariance. Invariance has been proposed as a fundamental and significant trait of biological perception towards object constancy [193–195]; i.e. the ability to conceive an object regardless of changes in size, position, illumination (3.3), pose, noise and rotation.

2. **Ease of use and efficiency.** In these early models, the processes have been clearly identified, introduced and enhanced while striking a balance between computational complexity, speed and biological-plausibility. For example, in FHLib multiple Gabor banks, previously introduced in HMAX are substituted with one spatial pyramid not only for computational speed but also because a hard-coded Gabor parameterisation scheme is incomplete and unrealistic. FHLib performs better than HMAX without sacrificing in biological-plausibility and is comparable to SIFT [196]. Also, GBVS introduces some enhancements over the original IKN model such as the incorporation of the motion, contrast and flicker features.

3. **Unbiased and bottom-up architecture.** All models perform bottom-up without any intentional influence. Top-down tasks are not considered in this work as they can specialise a methodology for a task but otherwise may underperform in other visual scenarios and situations. It is also difficult to estimate the influence and breadth of top-down biological processes with inconclusive physiological evidence, which could lead to ambiguous hypotheses. Therefore, a crucial guideline for this work has

been the effort to generalise and appropriately enhance the existing models in a biologically-inspired way.

4. **Compatibility.** These models are available for research purposes and have been developed in a similar framework using the same interface (MATLAB, C/C++) thus making their understanding efficient and integration fast, seamless and unified.

The visual attention models IKN (4.1.3) and GBVS (4.1.4) have been analysed in detail. However, these models suffer from certain identified drawbacks:

- The summation of dissimilar features (intensity, colour, orientation etc) into one holistic map via normalisation is uncertain to exist in biology and provides an ambiguous result. Similarly, the relationship between features has not been clarified or standardised in order to follow a predicted pattern given a particular task. For example, in GBVS, all features weights are by default set to 1 but the priority of one feature over another has an undefined value and therefore weight. It is very likely that the association of features for different objects and scenes should vary as well.

- As seen throughout chapter 2, neurons do not behave as “charging capacitors”, instead, they exhibit an all-or-nothing behaviour. This is in contrast to claims in [156], [197] where a winner-take-all strategy is adopted according to “charging capacitor” neurons. A similar technique is indirectly followed with weight attraction in GBVS, which in addition lacks accuracy since weight attraction (or activation) maps result in very broad areas within the image.

- The lateral inhibition mechanism, known to exist during the dorsal process (3.2), has a vague implementation.

- There is no evidence that the algorithms enhanced the overall performance of applications.

In turn, HMAX (4.2.3) and FHLib (4.2.4) do not fully capture the biological mechanisms behind object constancy and have also a number of major disadvantages that prohibit their performance:

- Feature extraction is a random process from across the image regardless of its content. In addition to this indiscriminate technique, there is no feature detection mechanism.

- The number of features in addition to the number of images that accurately defines and represents an object is always unknown. Moreover, there is no feature reduction method, especially against repeated features that may have occurred due to the random nature of the extraction process.

- Rotation invariance is absent and so an object aligned at different orientation angles will produce different results, some being significantly worse compared to the baseline.

- Colour recognition is also absent and both of the recognition models only investigate the morphological similarity between objects, essentially, neglecting important spectral information.

- Gabor filter parameterisation for both models follows empirical observations and does not offer any scientific insight on the mechanisms of simple and complex cells. Moreover, it can be proven that the one-for-all Gabor parameterisation strategy employed is inefficient, impractical and maladaptive.

- Given that Gabor parameterisation lacks a methodical approach, texture features cannot be fully realised and represented. Therefore, important textural information from scenes and objects is neglected.

- The classification part of the algorithm only uses a single “conventional” linear SVM, necessitating experimentation under different classification schema.

- FHLib had only been tested against one dataset.

The following sections of this chapter are devoted in addressing the drawbacks identified. More specifically, all image datasets used in this work are first presented in section 5.2. Subsequently in 5.4, the rotation invariance problem is tackled, followed by section 6.1.1 in which saliency is examined extensively both as part of the initial cooperation between visual streams and as a feature extraction method. In 6.2, colour features are introduced alongside the existing morphological and issues of illumination invariance are analysed. Finally, in section 7.2, methodologies on Gabor parameterisation and texture recognition are introduced and compared.

All experiments were conducted with MATLAB and where appropriate heavy computational load was passed to mex-files in a C/C++ environment. More information on MATLAB program codes is provided under each individual topic. Classification accuracy throughout this work is defined as:

$$A = \frac{S}{N} \cdot 100 \qquad (5\text{-}1)$$

This is the percentage of the test samples correctly classified *(S)* over the total number of samples in each classification experiment *(N)*.

5.2 Image Datasets

5.2.1 Video Sequences

The three different videos utilised in this work are each several minutes long and are split into thousands of frames. The “road” video sequence was captured outside Cranfield University in Shrivenham, the “exit from building” was taken outside a building on-site and the “surveillance camera” was directly taken from the “Performance Evaluation of Tracking and Surveillance” (PETS) 2009 benchmark data, University of Reading [198]. All sequences contain four object categories, background, bikes, cars, and people (For example frames see Figure 5-1). Under the background category falls every object observed in the frames of the sequences apart from the other three aforementioned classes. More specifically the sequences were designed so that:

1. All contain at least two of the four object categories.
2. Objects maintain some distance from the camera.
3. There are ideal weather conditions.
4. There is sufficient illumination (daylight).
5. Interesting objects for the most part of the videos are non-occluded.
6. Camera views are static.

The video sequences have not been an extensive part of this work and have been used only in section 5.3 to illustrate multiple salient objects recognition and to measure the continuous recognition responses of the algorithm.

Figure 5-1: Example frames representing each of the three video sequences.

5.2.2 Static object datasets

Under the static object category, datasets are separated in 3 thematic categories: multi-class, colour and texture, following the requirements of the experiments in each section of this chapter. Multi-class datasets are generalised without significant interclass patterns while classes in colour datasets share similar spectral information and in cases share morphological information. Texture datasets strictly portray patterns in texture.

The Cranfield University Uncluttered Dataset (CUUD) is a small multi-class dataset that consists of four categories of vehicle images and these are: airplanes, bikes, cars and tanks. Images have been collected from the internet and the publicly available database INRIA. Each image contains only a particular vehicle without any clutter or obscurances (Figure 5-2). The images are of varying aspect ratios and their resolution is always higher than a minimum of 240 pixels for their shortest edge. Objects are in varying directions and portray some variation in spatial position.

The Cranfield University Cluttered Dataset (CUCD) has also been partly assembled from the internet (publicly available database INRIA) and in part from images collected across the campus. All images in the dataset contain background clutter and belong to four categories: background, bikes, cars, and people. The background category shows a great variability of information, i.e. buildings, roads, trees etc. The purpose of the background category of images is to disassociate possible background feature fusion with the other 3 classes. The people's category is the only category of non-rigid objects, therefore, within this category pose and position varies significantly. Another difference with respect to CUUD is that in an image there may be more than one object (of the same category) present. Similarly to CUUD, the images are of varying aspect ratios and their resolutions are always higher than a minimum of 240 pixels for their shortest edge (Figure 5-3).

Figure 5-2. Example images from the CUUD vehicle classes.

Figure 5-3: Example images from the CUCD dataset.

The original “butterflies” colour database from the University of Illinois at Urbana – Champaign (UIUC), consists of seven classes of butterflies and in total contains 619 images for all classes [199]. The dataset classes “Monarch 1” and “Monarch 2” were merged into one class as their similarity would not benefit tests on colour recognition. Moreover, the class “black swallowtail” was enriched with more representative images from the internet since the initial number of images was low (Figure 5-4). The “birds” colour database also from UIUC, consists of six different classes of birds and in total contains 600 images [200]. This dataset is used unaltered and some examples can be seen in Figure 5-5.

The “cats” colour dataset is another newly introduced dataset created for this work and more specifically for the round of experiments in section 6.2. The dataset was assembled from the public domain “Imagenet”. It contains in total 5381 images separated into six classes of cats namely, “cheetah”, “lioness”, “panther”, “siamese”, “snowleopard” and “tiger”. The choice of these particular species of cats is significant since they all have a unique texture-colour relationship. For example, tigers have a golden fur with stripes, cheetah gold furs with spots and snowleopards white/grey coloured furs with spots. Siamese cats and panthers are reversely coloured (white-black) and are also characterised by the lack of texture (spots or stripes) in their fur. All cats are illustrated at various poses, image locations, environments and illumination conditions (Figure 5-6).

Figure 5-4: Some examples of the UIUC “butterflies” dataset.

Figure 5-5: Some examples of the UIUC "birds" image dataset.

Figure 5-6: Some examples of the Cranfield University "cats" image dataset.

Another dataset from the UIUC [201] was created for texture recognition tasks. It originally consisted of 25 classes each representing different texture materials such as wood, marble, water etc. Each class has 40 greyscale images at 640x480 pixels. In this thesis (section 7.2), the first 10 classes are used (for faster computation) and this part of the dataset is named as "TX10" (Figure 5-7)

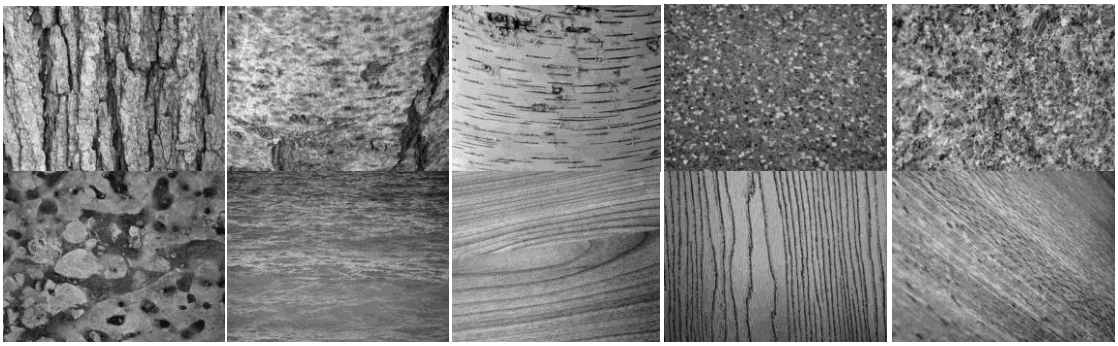


Figure 5-7: Some examples for each of the 10 classes in the "TX10" image database from UIUC.

The 10 multiclass dataset includes three categories from CUCD (background is excluded) and expands on a variety of classes. All images were obtained from the internet and obey the selection criteria set already in CUCD. Similarly, the 25 multiclass dataset expands even further while maintaining the original classes from the CUCD and 10 class datasets. The 10 and 25 class datasets as a requirement have a higher resolution equal or above 240 pixels (Figure 5-8 and Figure 5-9).

Figure 5-8: Some examples from the 10 class Cranfield University dataset.

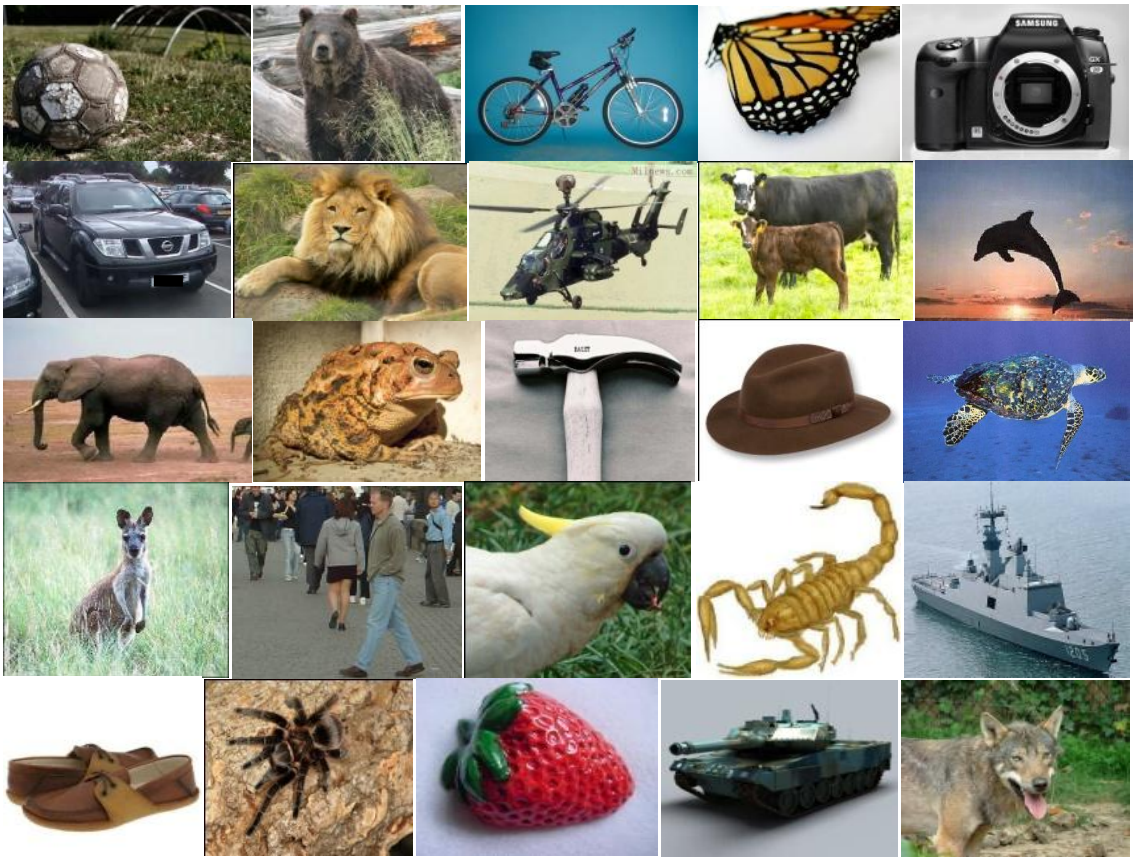


Figure 5-9: Some examples from the 25 class Cranfield University dataset.

The multiclass image dataset Caltech 101 [202] consists of 101 different object categories including one for backgrounds. A total of 9197 images on various spatial poses include unobstructed objects mostly centred in the foreground with both cluttered and uncluttered background environments. All images have been taken at different aspect ratios and are always higher than a minimum of 200 pixels for their shortest edge (Figure 5-10).

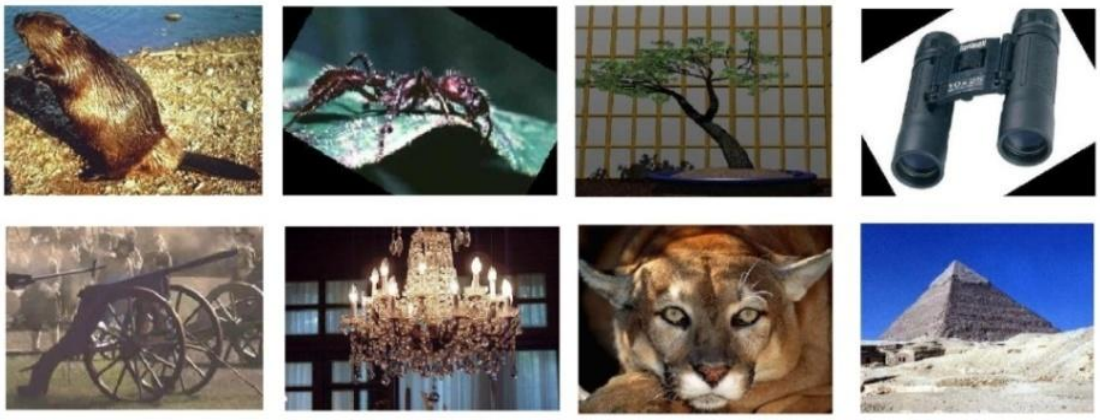


Figure 5-10: Some examples of classes from the 101 Caltech dataset.

5.2.3 Colour constancy datasets

The colour constancy datasets of this section are used in section 7.1.2 in order to train an SVM classifier that given an image's semantic information, can decide the appropriate colour constancy method.

The "Mondrian" image dataset [96], consists of 310 Mondrian-like images with different edge properties, texture, shadow gradients and contrast to closely simulate real world illumination transitions. Some examples of this dataset are illustrated in Figure 5-11. This dataset also contains ground truth data over the *R, G* and *B* illuminants and is used only in the experiments of section 7.1.2 along with the naturalistic "Barcelona" dataset described below.

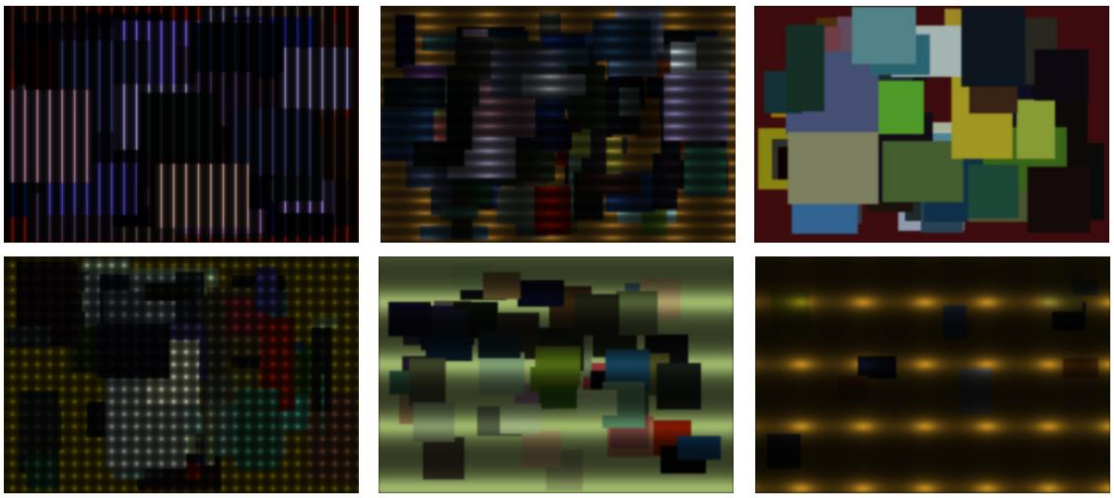


Figure 5-11: Examples of the Mondrian-like dataset used for the colour constancy fusion approach.

The “Barcelona” dataset (Universitat Autonoma de Barcelona) [203] [204] contains several different thematic sub-categories each containing various numbers of images and is available in the CIE1931XYZ or LMS colour spaces. It is a device–independent dataset captured with a calibrated camera and all images are assumed to be taken under the D65 illuminant. Each image has a grey ball (RAL 7012) at a fixed distance on the bottom-left corner of each image, serving as a uniform spectral reflectance area and Lambertian surface (i.e. isotropic reflectance regardless of observer angle). The creators provide the ground truth data for every pixel of an image and also their RGB images. In this thesis, the “naturalistic_01” and “naturalistic_04” are merged and hereafter referred as the “Barcelona” dataset (Figure 5-12).

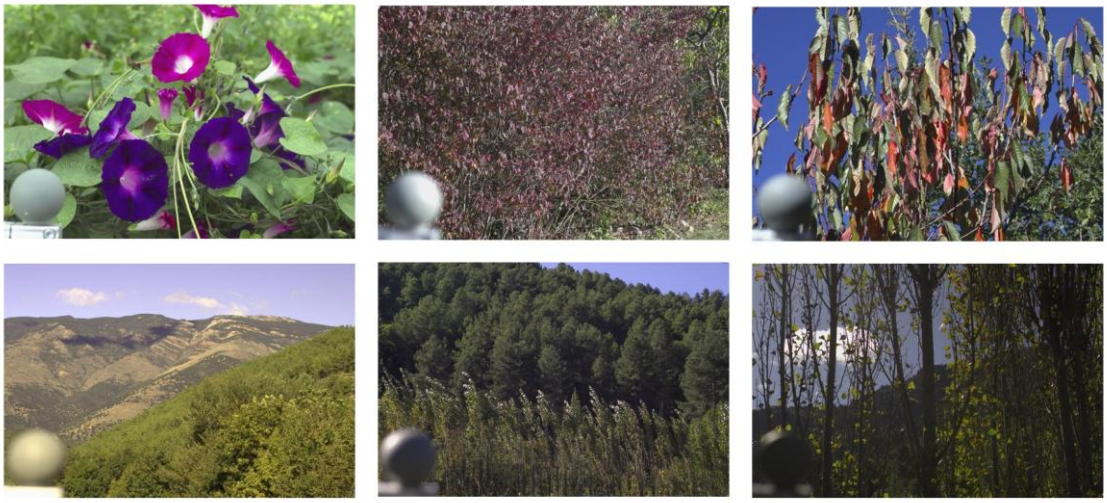


Figure 5-12: Examples of the “Barcelona” dataset of natural scene images.

To take advantage of the CIE XYZ “Barcelona” dataset the following procedure in MATLAB was followed:

1. Extract median values for red, green and blue values for the grey ball area using ground truth images for every scene.

2. Load RGB images and apply procedure in section 3.3.2.

3. Compare performance of each of the colour constancy methods against values from step 1.

Finally, in section 7.1.2 a merged version of the “Mondrian” and “Barcelona” datasets is also used and referred as the “Mix” dataset. It is simply an emergent dataset of both datasets within the same vector space.

5.3 Proto-objects

The main goal of this particular section is to combine visual attention and object recognition, as a first step of communication between the two visual streams. In the last decade, several studies have proposed methods towards a unified approach between attention and recognition with varying degrees of biological plausibility, complexity and success. For example, Walther's model [197] introduces the notion of "proto-objects" i.e. a sequential salient method of learning sub-regions in a visual scene regardless of content. Regions of images are prioritised bottom-up and attended according to their saliency impact using IKN. These ROI are subsequently processed with HMAX which performs the recognition aspect of the algorithm. In Walther's model, the intrinsic properties of objects are not examined or the architecture of the two models changed. Instead, this model concentrates on the recognition of areas in a visual scene, lacking a thorough experimental analysis and validation. Likewise in [205], contrast, centre-surround histograms and colour distribution saliency maps are employed to identify salient objects with prior knowledge. Intrinsic properties of objects in this approach are also neglected in an attempt to segment salient objects from their scene under a method which arguably has less biological realism. In [206], the proposed method uses Harris corner detection and salient points obtained from Haar wavelet transforms to learn objects aided from statistical methods such as Principal Component Analysis for dimensionality reduction and K-means for clustering. Given the small dataset, lack of comparison and superficial experimentation it is difficult to conclude whether this model has been improved under the proposed methodology. Recent work in [207] unified IKN-based saliency maps with Bayesian probability theory for top-down visual search tasks, in other words predicting the location of salient objects and features given prior knowledge. Their method is compared against the well-known model SIFT and HMAX, portraying ambiguous and dataset-dependent behaviour. Lastly in [208], authors use an IKN-based saliency model with HMAX for scene classification, reporting a slightly inferior object recognition performance for biologically inspired features against the statistical SIFT. Nevertheless, it is hard to justify this conclusion on a single dataset and more work is needed to confirm this result.

The procedure here resembles the approach in [197]. The first (proto) interesting objects or regions in an image are sequentially identified without any prior information about the scene or intentional influence. Another objective of this section, in contrast to work in [189], [197], is to analyse the recognition performance against an increasing number of features (more informative) and an increasing number of training images (more diverse). The association between them should provide a guideline for choosing the number of features and images in subsequent sections.

5.3.1 Method

As soon as visual stimuli reach the V1 area in the occipital lobe of the brain, the two visual paths simultaneously process in parallel, the “where/how” and the ”what” information while sharing interconnections for cooperation. Amongst other tasks, the dorsal processing stream is responsible for handling visual attention while the ventral processing stream deals with object recognitions and associations (sections 4.1, 4.2). In the first instance this biological behaviour can be simulated directly by using GBVS and FHLib. This unification leads to a completely bottom-up approach and it is a simulation of the first response of biological visual perception.

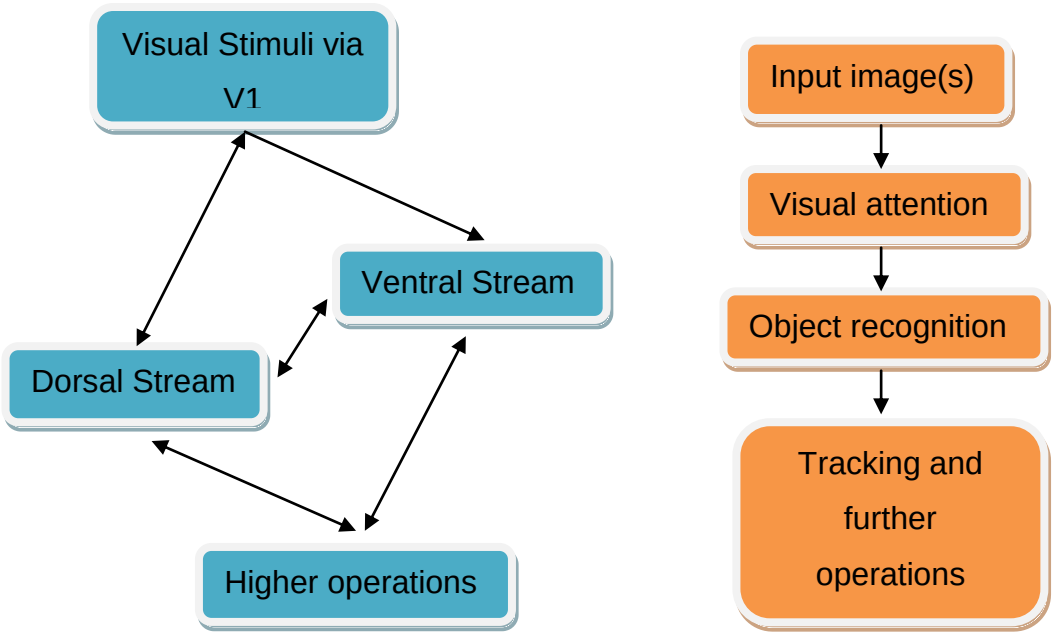


Figure 5-13: Mimicking biological behaviour. Left diagram of boxes shows a general operation layout of the visual cortical pathways. Right diagram illustrates the procedures followed in this section to mimic biological behaviour

Processing the input images (or frames of a video) is done according to the simple procedure:

1. Input image processed from GBVS.
2. Salient or interesting objects are isolated and then ROI become the testing images for a pre-trained FHLib setup which returns the recognition results.

5.3.2 Experiments Setup

The algorithm in GBVS does not provide an optimisation mechanism for evaluating its parameters. So throughout the experiments in this section, GBVS parameters are kept at default (Appendix B). Similarly, in FHLib the setup has being chosen at default while its Gabor parameterisation is examined more closely in section 7.2 of this thesis. The original MATLAB code has been modified for both implementations GBVS and FHLib in order to accommodate the concept in Figure 5-13. For this set of experiments, the CUCD dataset has been used to train the Support Vector Machine classifier.

SVMs implement a simple idea, to view input data as two sets of pattern vectors which are then mapped in a high dimensional feature space where a separating hyperplane can be constructed. The separating hyperplane's purpose is to maximise the margin between the two sets of vectors. Suppose a classification problem where sets training samples $x_1 \ldots x_n$ are assigned to either class ω_1 or ω_2 then by considering the equation for a linear discriminant function [209]:

$$g(x) = w^T x + w_0 \tag{5-2}$$

where w^T is the weight vector and w_0 is the threshold value, the following decision rule needs to satisfied [209]:

$$w^T x + w_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow x \in \begin{cases} \omega_1 \text{ with value } y_i = +1 \\ \omega_2 \text{ with value } y_i = -1 \end{cases} \tag{5-3}$$

Thus, all training points are correctly classified if:

$$y_i(w^T x + w_0) > 0 \text{ for all } i \tag{5-4}$$

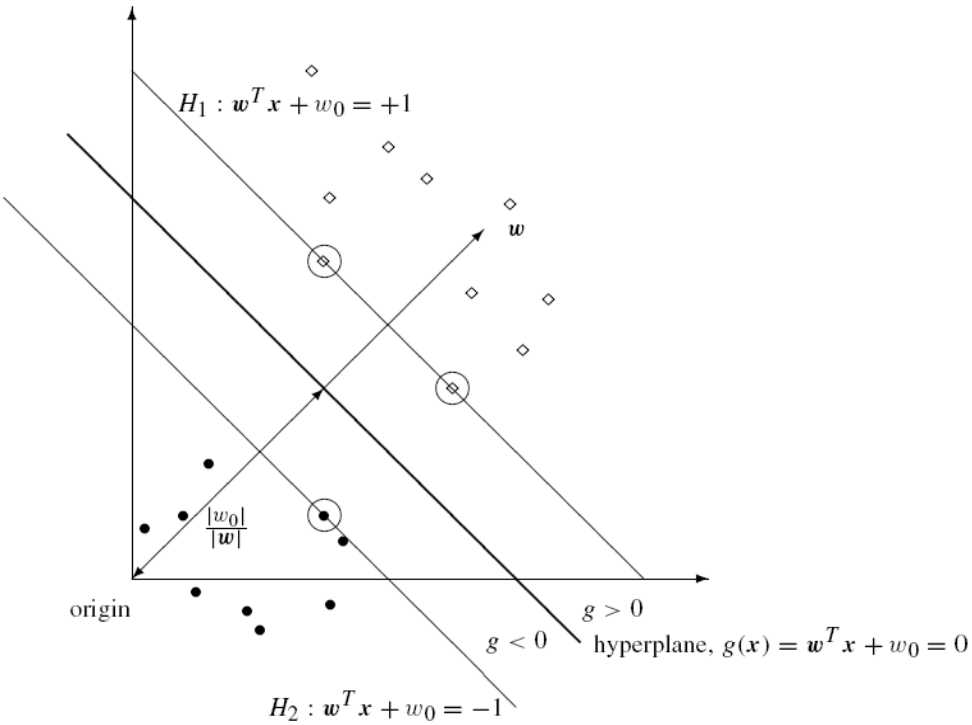


Figure 5-14: An example of the SVM operation. H1 and H2 are canonical hyperplanes. The support vectors are the points inside the rings [209]**.**

By applying the kernel trick, a non-linear SVM classifier can be created. Furthermore, the problem of a multiclass SVM is solved by reducing to multiple binary problems. Two common methods exist to build multiple classifiers:

1. One versus all, where classification relies on the winner-takes-all strategy

2. One versus one, where classification relies on the max-wins strategy

SVM require a considerable amount of time to calculate with an increasing number of classes and dimensions.

In each run the total number of extracted features and the total number of training images is varied to obtain different FHLib codebooks. The numbers of features have been chosen as 1000, 3000, 5000, 8000, 10000 and 15000 and have been varied against 120, 200, 400, 600 and 1000 training images/category. The choice of the number of features is comparable to the number selection of features in [189] but the number of training images was considerably increased from a maximum of 30 training images/category in [189] to a minimum of 120.

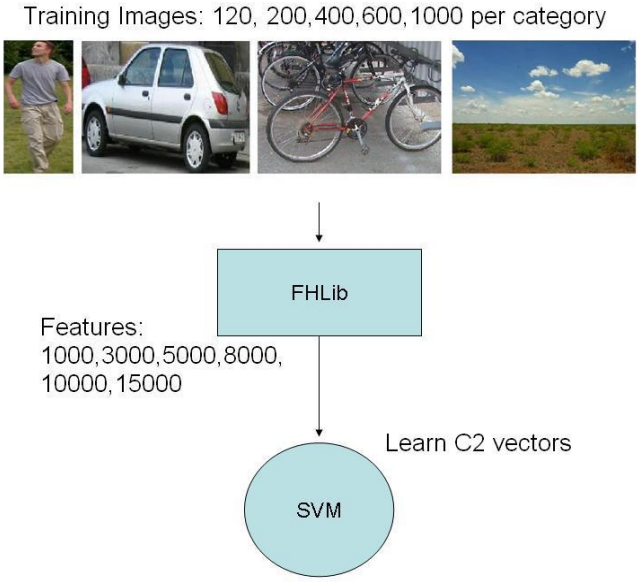


Figure 5-15: An illustration of the training procedure in this section.

At the testing stage, the video frames of the three sequences (5.2.1) are used as inputs for the GBVS MATLAB code. A weakness of visual attention in GBVS as explained in the introductory section 5.1 is that feature choice is undefined and so in a given frame the spectral or intensity information across the scene would provide misleading information. Therefore, the selected features were motion and flicker, since the targets in the videos required for recognition are mostly in motion (Figure 5-16, Figure 5-17, Figure 5-18).

Figure 5-16: Top row shows video frames of the road sequence. Bottom row the same video frames produced via GBVS saliency maps detecting the targets of motion.

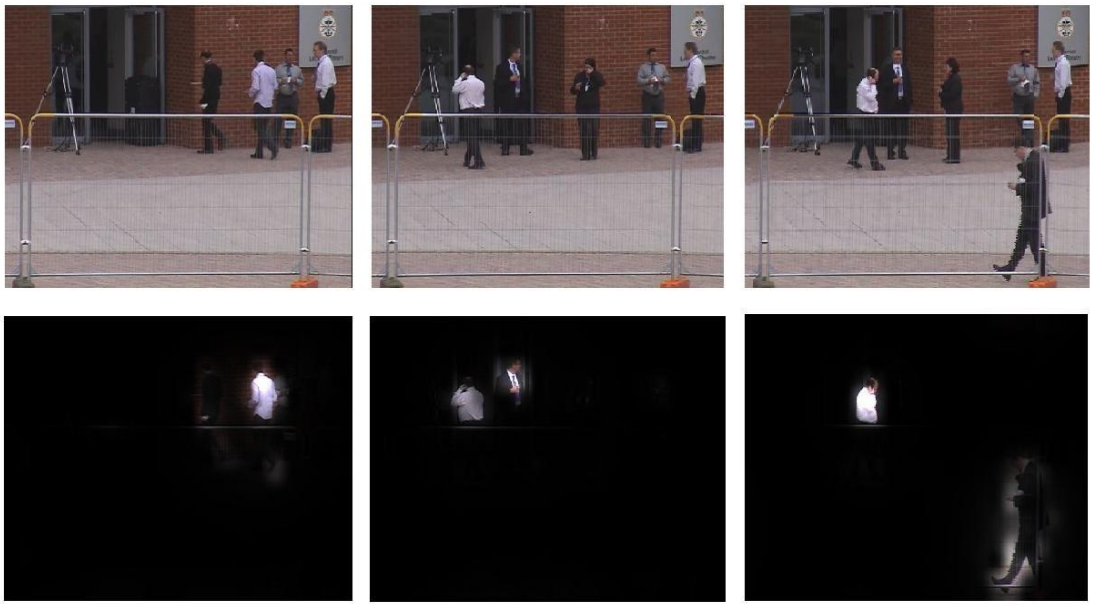


Figure 5-17: Top row shows video frames of the building exit sequence. Bottom row the same video frames produced via GBVS saliency maps detecting the targets of motion

Figure 5-18: Top row shows video frames of the surveillance camera sequence. Bottom row the same video frames produced via GBVS saliency maps detecting the targets of motion

The shaded regions of the frames are for illustrative purposes and highlight salient objects projected through saliency maps. Shaded projections of targets however, would not be appropriate inputs for FHLib since spectral and spatial information is distorted. Consequently, the salience map acts as a topographic

map providing the coordinates at which FHLib can perform recognition as shown in Figure 5-19.

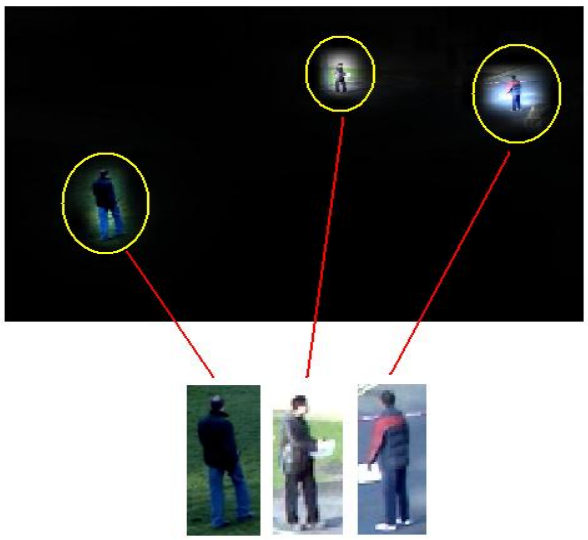


Figure 5-19: An example of three salient ROI providing coordinates for FHLib.

There are some motion problems as the camera accidentally shakes at some instances and creates false motion across the scene (Figure 5-20). In addition, at instances where targets remain immobile (regardless of whether they were mobile for some time or not), GBVS cannot continuously detect them, exposing another weakness of this visual attention approach (Figure 5-21 and Figure 5-22).

Figure 5-20: A video frame from the building exit video sequence, after the camera has just moved by a fraction. Having selected the motion and flicker features falsely indicates movement across the scene.

Figure 5-21: An example where the GBVS motion-flicker feature stops detecting immobile targets in the surveillance camera video. Top row shows original video frames.

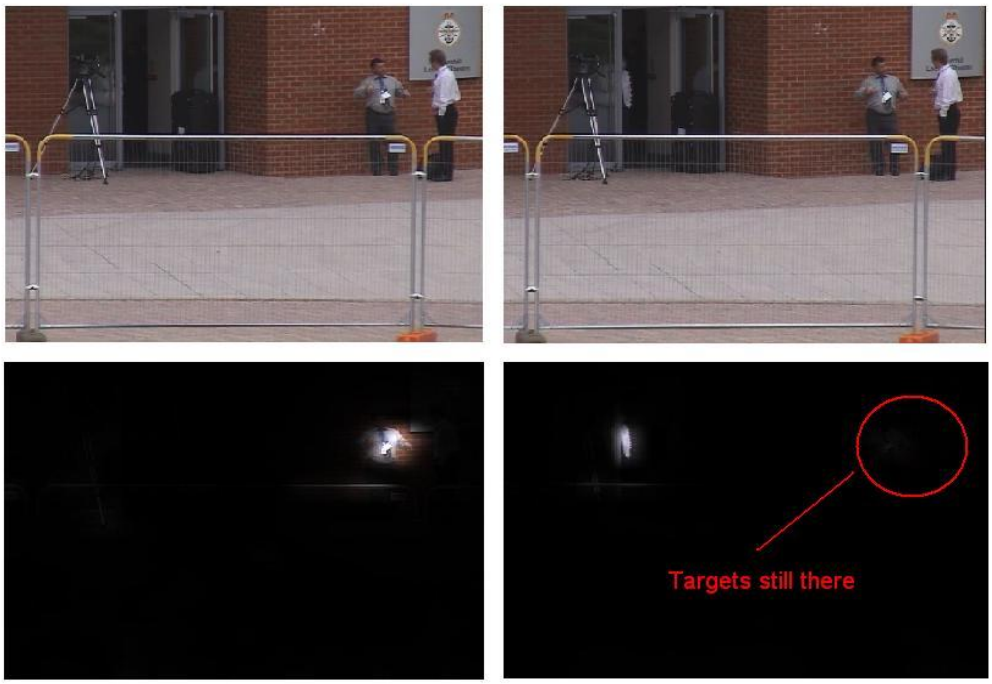


Figure 5-22: An example where the GBVS motion-flicker feature stops detecting immobile targets in the building exit video. Top row shows original video frames.

The *C2* vectors for each object are obtained using the Gabor filter procedure (4.2.4) and passed onto a multi-class linear SVM classifier under a one-against-one decomposition mode for the classification. For both the training and classification stages, note that the input to the SVM is the *C2* vectors and not the raw image data. Prior to classification, and as in [189], the image data is sphered i.e. the mean and variance of each dimension are normalised to zero and one respectively. The majority-voting method has been adopted for the classification step and the recognised ROI are portrayed over the salient regions as the bottom row of results (Figure 5-23). Some examples of the classification approach of this section across various frames is shown in Figure 5-24. The classification setup was set as default from the original algorithm and in later sections of this thesis it is addressed in more detail. In fact, SVM is a non-biological classifier but as a concept of comparing classes in a one versus one and one versus all method, is hypothesised to exist in task-specific neuronal circuits in the prefrontal cortex [188].

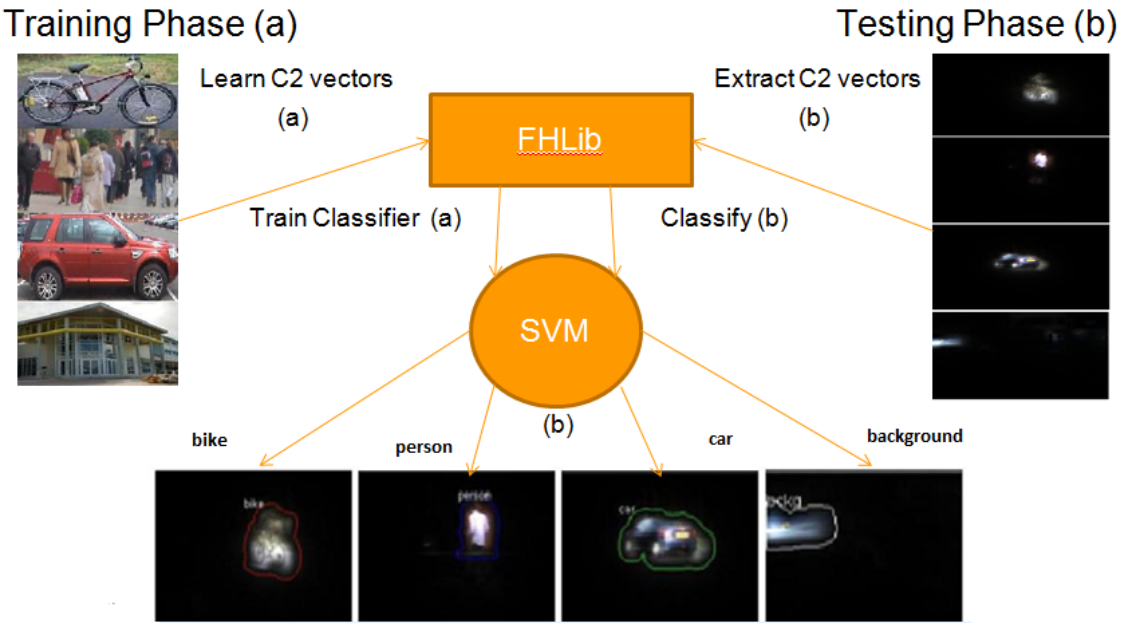


Figure 5-23: The training path (a) prepares the FHLib's feature library while training the classifier. As the ROI are fed, their *C2* vector responses are compared against the codebook and then classified to find the best category match. The bottom row of images shows labelled ROI correctly classified.

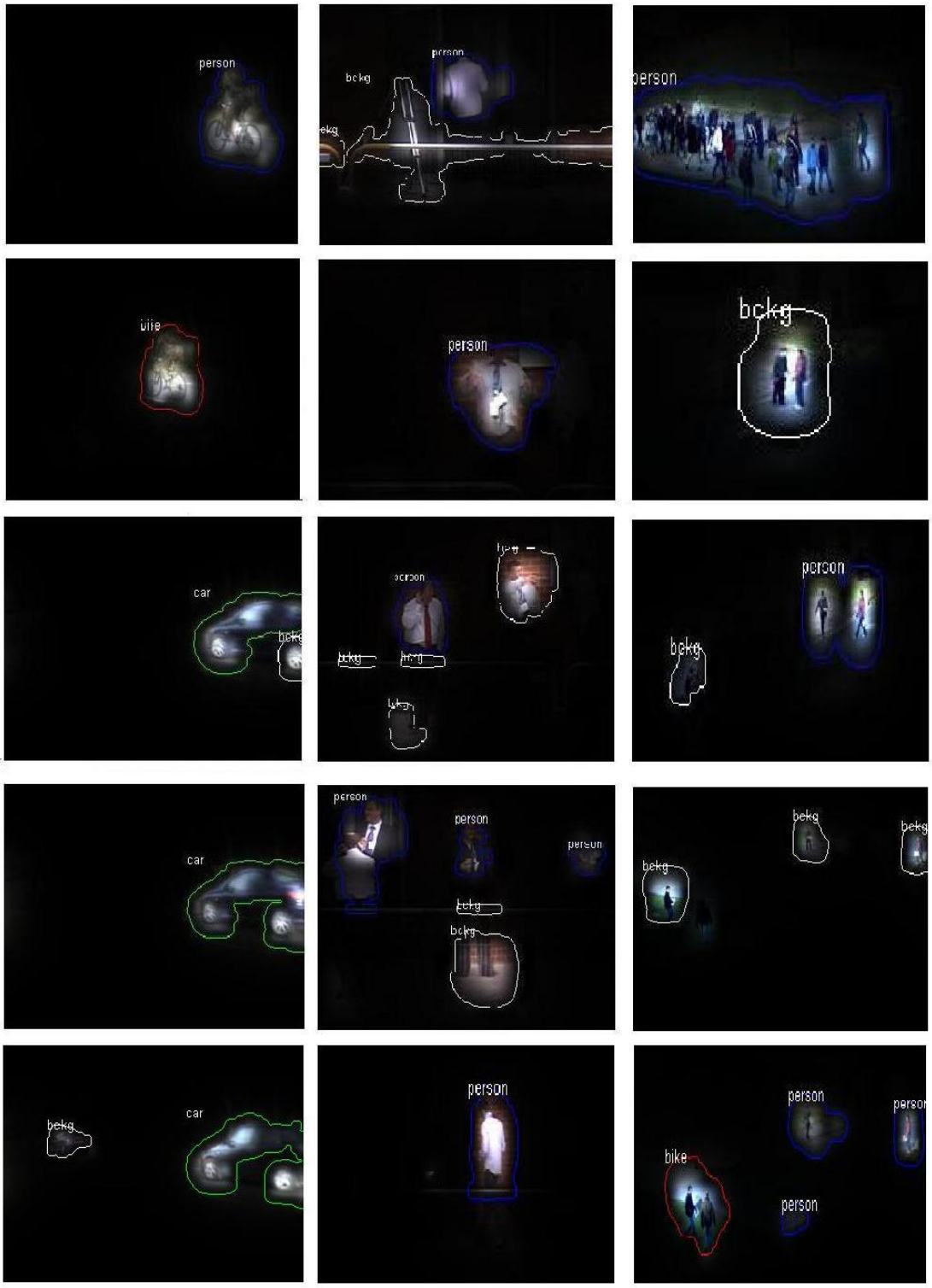


Figure 5-24: Some examples of classification in frames from the three video sequences (left column to right column - Road video, exit from building, Surveillance camera videos). These results were obtained from 150 training images/category and 1000 FHLib stored features.

Table 5-1 below, shows the classification accuracy of *C2* features (NOF) used for classification, together with the number of training images (NOTI). The classification scores are the averaged accuracies over three test datasets employed in this work. It is quite clear that the classification accuracy is weakly dependent on the NOF, but it is significantly improved for a medium range of NOTI. Furthermore, after 3 independent runs the accuracies as presented in Table 5-1 do not vary more than +/- 2% on average. A direct comparison of the present result with respect to previous work indicates an agreement to FHLib data and the results have shown a much improved classification accuracy when the NOTI is increased to about 150 per category. Importantly, this table also shows that a computer can do what a human does beyond a percentage of chance.

Classification Accuracy (%)	N.O.T.I 120	N.O.T.I 200	N.O.T.I 400	N.O.T.I 600	N.O.T.I 1000
N.O.F 1000	55	61.89	67.87	77.74	62.27
N.O.F 3000	54.83	60.77	70.27	69.45	58.70
N.O.F 5000	53.71	62.14	70.44	70.94	71.05
N.O.F 8000	55.11	64.29	70.93	73.15	62.10
N.O.F 10000	55.41	63.40	70.34	73.21	68.57
N.O.F 15000	55.30	61.31	69.17	74.37	69.32

Table 5-1: The average classification accuracies of the SVM over the three sets of video test streams as functions of number of C2 features (NOF) employed in the classification, and number of training images (NOTI) used for training the classifier.

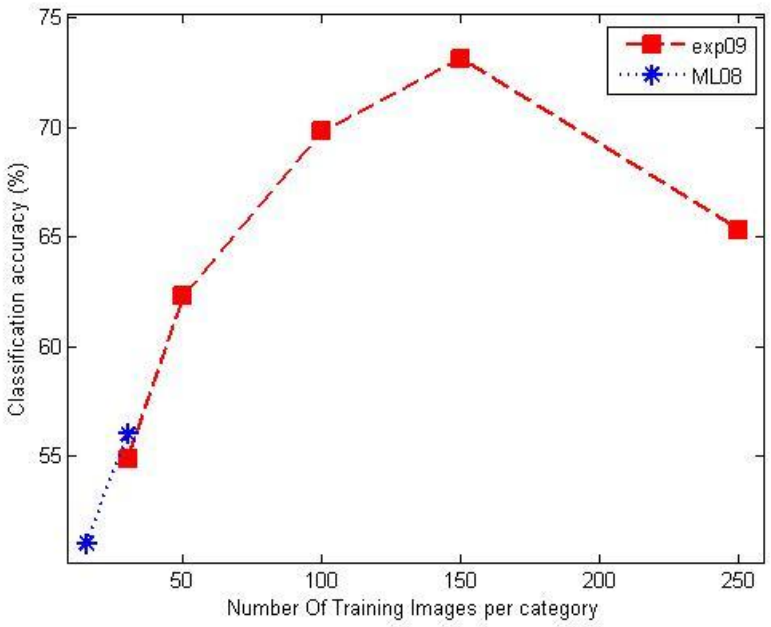


Figure 5-25: The averaged classification accuracies for the three datasets employed in this study. The drop observed at 250 images per category could be due to changes introduced with the added features in the dataset. The results show agreement with previous work shown in blue [189]**.**

In Figure 5-25, an obvious decrease in classification accuracy can be observed from the preliminary experiments at 250 NOTI. The reason seems to be that as the number of training images is increased against a given number of randomly selected features the data becomes very sparse, reducing the overall quality of feature data. For example, for 1000 features at 50 NOTI (i.e. a total of 200 training images for four categories) means 5 features per training image are stored in the FHLib's codebook. For 1000 features and 1000 images in total, the codebook contains one feature per image. This can also be realised from experiments in [189] where FHLib is tested against the 101-Object categories Caltech University dataset where for a total of 3000 training images a 12000 feature codebook is first created (4 features per image) and from this pool of feature data, a further feature selection is performed to refine a category's feature representation. Thus, it seems there is a balance in the number of features versus the number of training images (i.e. between information and diversity) and its evaluation can be critical for classification accuracy. Such optimisation is dependent on the classification task and dataset. In following sections, the choice was made to over-represent objects by selecting an abundant number of features rather than under-representing which might lead to poor performance. To balance the negative effects of over-representation of objects, a purification mechanism is required to eliminate unnecessary features.

This comes at a computational cost but without a measure of accurate description it is difficult to estimate feature number versus number of images.

5.3.3 Section conclusions

Working towards a unified model for biologically-inspired computer vision in this preliminary experiments section has exposed a number of issues which are the subject of subsequent analysis in this thesis:

- Parameterisation needs to be adaptive.
- Scale and position invariance alone are not enough for object constancy.
- The random extraction of features from the recognition part of the algorithm requires refinement against background clutter or “noise”.
- The visual salience model needs to be refined.
- In biological visual cognition, humans do not only use spatial information but a number of different patterns emerging through the use of both spectral and spatial information.
- More datasets and classification schemes are required.
- A biological-like classification schema along with a comparison against other classifiers should be employed to validate results and improve classification accuracies.

5.4 Rotation invariance

Physiological experiments have proved the existence of a fast and efficient rotation-invariant mechanism in humans [210], [211] for simple objects and classes such as cars but not complex patterns. Nevertheless, there is still uncertainty on how such a feat is practically achieved. The findings in [211] lead the authors to conclude that given the “ultra” fast (140-260 ms) and high (above 80% over all rotation angles) recognition rates on both pre-learned and novel images, biological rotation invariance is not a mental process. This complements the analysis from a number of studies and experiments in [210], which in addition concludes that some mental processing may be involved especially when the orientation of an object needs to be determined.

This section introduces rotation invariance as an additional object constancy element to the existing shift and scale invariance properties in biologically-inspired object recognition. Generally, a key ability of biological vision is object constancy and all biologically-inspired models in some way are attempting to replicate its mechanisms. For example, the Neocognitron has invariance to shift, Convolution Neural Networks limited invariance to shift, size and distortions, SIFT and SURF [212] partial invariance to shift, size, rotation and illumination only available for redetection of the same components and HMAX/FHLib invariance to shift and size. Experiments in [213] show that HMAX

portrays moderate position and size invariance, and poor rotation invariance. Just 15 degrees of rotation causes a nearly 30% decrease in recognition performance [213]. Likewise, results for FHLib show poor behaviour and are presented analytically in section 5.4.1.1.

Various methodologies in the past have been employed to tackle rotation invariance problems. SIFT originally utilised image gradients to assign a canonical orientation and orientation alignment to match its descriptors [196]. Other SIFT variants like PCA-SIFT [214] and Gradient location-orientation histogram (GLOH) [215] follow similar approaches. A slightly different approach of image gradients within concentric rings is followed in RIFT [199]. SURF follows orientation assignment by first applying Haar wavelets and then summing the responses within a certain radius from an interest point to obtain a single vector [212]. Histogram of Oriented Gradients (HOG) descriptors use image gradients with spatially overlapping blocks (cells) of histograms [216]. A thorough comparison between orientation assignment methods is presented in [217]. After edge detection with Gaussian derivative filters three different techniques are examined, a gradient orientation on the centre of a pixel neighbourhood, peak of the orientation histogram and eigenvector orientation of the second moment matrix.

As explained above, the physiological experiments have proved the existence of a biological rotation invariance mechanism and this topic remains largely unexplored for biologically inspired object recognition. This section and its experiments attempt to initiate such research.

5.4.1 Rotation Invariance methods

5.4.1.1 Object orientation alignment

In this section the major changes over the original MFHLib are:

- The inclusion of a rotation invariance mechanism at the *S1* layer.
- The number of different orientations is increased to 18, each at steps of 10 degrees between 0^o and 360^o.

Following a histogram approach the identification of the dominant orientation for a single feature of an object is found holistically. The MFHLib algorithm at the *S1* layer has Gabor filter batteries tuned at different orientations and so yields maximum responses in each i.e. detecting the edges. For each orientation a histogram is created with a number of bins controlled by the difference between minimum and maximum values. The mean number of values is then found and the number of *S1* units above the mean indicate how many units have responded to a particular orientation. The maximum number of detected values

shows which orientation responded maximally. For this reason, the number of orientations was increased in order to accommodate more precise orientation information.

The dominant orientation of an object is therefore the sum of values above the mean. When the dominant orientation (*D*) is obtained at the *S1* layer, then if $D \neq 0$, let α equal the angle of rotation required to rotate the image to 0 degrees so that:

$$R = D - a = 0^o \quad \textbf{(5-5)}$$

During the training phase, the dominant orientation of an object in a training image is first identified. The chosen reference orientation here is at 0 degrees (any value from 0^o to 360^o could serve as reference) and if the object in an image is oriented at a different angle then the model simply rotates it to 0 degrees. In this way, the *S2* templates are aligned to the reference orientation. At the testing phase, the same approach brings all testing images to the same reference orientation.

Limiting geometric invariance above the *S2* layer has been attained in a different method to [189]. To preserve spatial representation of an object in an image, one side (the shortest) is always scaled to 140 pixels while the larger side has a length equivalent to the ratio between the two original sides, in other words the aspect ratio is maintained in an identical way to FHLib. However, during the training stage, as the coordinates of the *S2* features are stored so do the spatial distances and sizes. This means that if a testing image has a different aspect ratio then the model should readjust the coordinates according to the aspect ratio of the testing image regardless if it is larger or smaller than the original. For example, assuming that a patch was extracted from image coordinates *x = 50, y =50* of an image 140x200 pixels in size, on a testing image of 140x140 pixels in size then the new coordinates should lie at *x = 50, y =35.* So, around an area of the new coordinates and at the range of scales chosen by the user (default 1 scale) the model is searching for the maximum value.

5.4.1.2 Local Feature rotation

The method described in the previous section is a naive approach for experimental reasons and bears no practical significance since it is expected to operate optimally in the absence of background information or with an efficient object segmentation mechanism. Consequently, a local feature rotation mechanism is presented here, applicable to a wider selection of datasets and scenarios.

With local feature rotation, dominant orientation assignment is unnecessary. Instead, *S2* features during the training stage are extracted as in FHLib (section 4.2.4). However, during execution of template matching over the *C1* layers of the training images, each *S2* template is rotated from 0-360 degrees in steps of 30 (12 distinct angles in total) and each feature gets 12 different RBF distance values. The maximum is selected to express the best match as a *C2* vector. A similar process occurs during testing as well.

An outline of the major algorithmic steps is given below:

1. Use parameterisation and training images from the image layer to the *S2* layer similar to section 4.2.4 using MFHLib.

2. Each *S2* template is then run over the *C1* layers of each training image and rotated from 0-360 degrees in steps of 30 degrees. Each set of results produces a single maximum response.

3. The maximum response from step 2 is stored as a *C2* vector value in the featurebook and used to train the SVM classifier.

4. During training, all *S2* templates are similarly to step 2 run over the testing image *C1* layer 0-360 degrees in steps of 30 degrees and each set of results produces a single maximum response.

5. All maximum responses are stored and used in the pre-trained SVM classifier to obtain test results.

5.4.2 Original FHLib at different rotations

In the absence of a segmentation mechanism which would limit or eliminate background edges and shapes being introduced, in both HMAX and FHLib, Gabor filters will indiscriminately detect edges from background clutter. Without a segmentation mechanism these unsupervised learning models are inevitably prone to shape distortions (Figure 5-26). Even if feature selection and refinement was employed as a post-processing measure it seems unlikely that such a method would produce far better results without a serious computational price or supervised intervention. In biological vision, primates combine their visual abilities to segment objects from cluttered environments for example, stereoscopic vision offers the perception of depth, consequently extract foreground objects and estimate distances. Similarly the perception of colour also plays an important role in understanding the spectral continuity of an area.

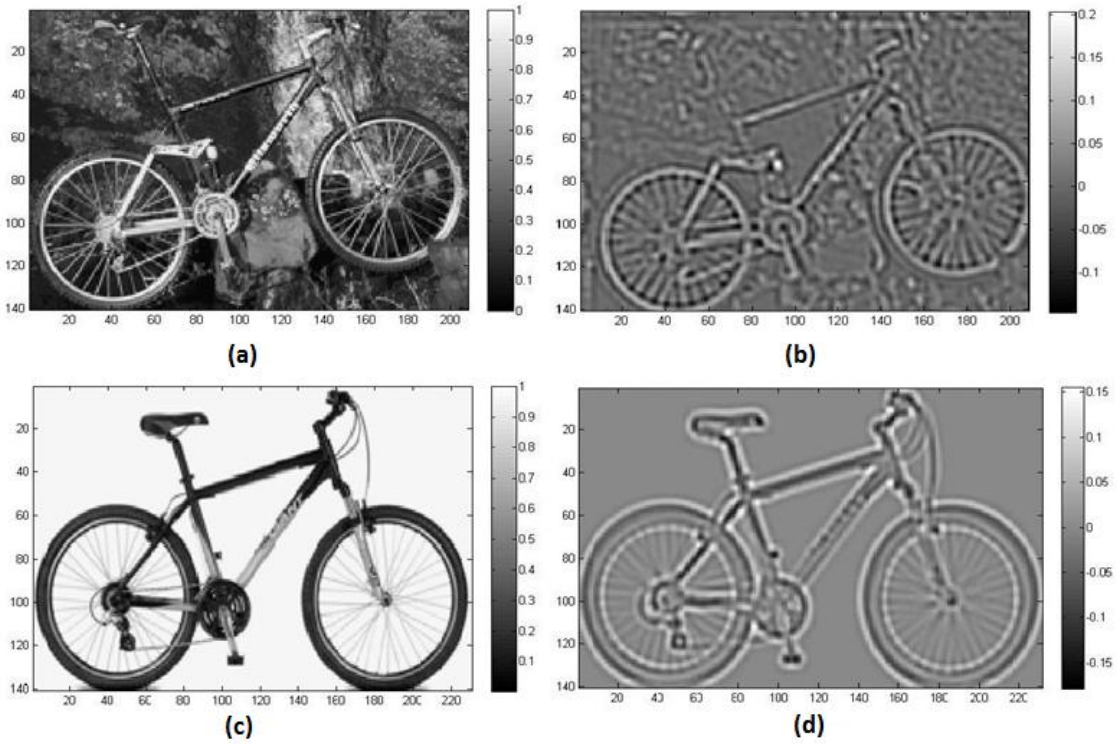


Figure 5-26: Cluttered vs. uncluttered background in similar images of bikes. (a) the input greyscale image of a bike with background clutter, (b) the Gabor filter output of (a) in 12 orientations, (c) the input grayscale image of a bike with uncluttered background, (d) the Gabor filter output of (c) in 12 orientations.

FHLib can perform at an average of 87% under the CUUD dataset (after 3 independent runs, 50 images per category for both training and testing). The *C2* vectors for each object are obtained using Gabor filters as described in section 4.2.4, and they are then passed into a multi-class SVM classifier under a linear one-against-one decomposition mode for classification. Note that the input of the SVM is the *C2* vectors and not the raw image data for both the training and classification stages. Similar to previous work [189] the image data is "sphered" i.e. the mean and variance of each dimension are normalised to zero and one respectively, prior to classification. The majority-voting method has been adopted for the classification step. For a general outline of the procedure refer to Figure 5-27. Finally, the Gabor parameters are set to default $\gamma = 0.3$, $\sigma = 4.3$ and $\lambda = 5.6$, while the inhibition parameter $h = 0.5$ and parameter t_s is set to 1. The interest of this work is not to show as in [218] where the classification accuracy is the highest but rather to illustrate the rotation invariance behaviour of FHLib. Therefore for all experiments the training dataset consisted of 50 images per category (200 in total) and 50 different testing images per category (200 in total).

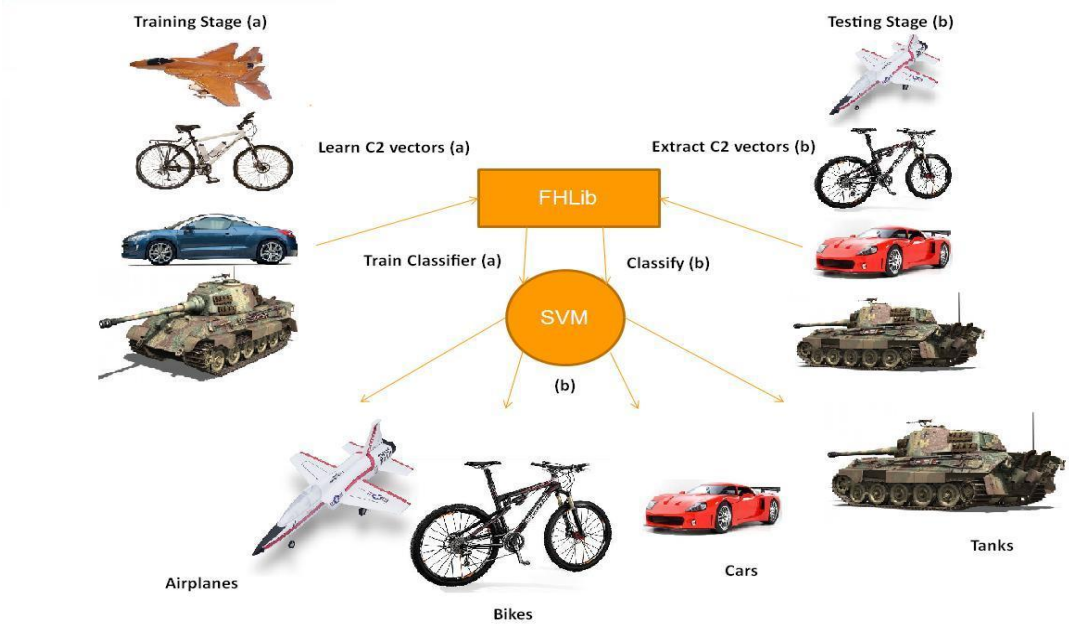


Figure 5-27: A general algorithm layout of the experiments in this subsection. In (a) during training phase, each class from the CUUD is fed into FHLib in order to train an SVM classifier. In (b) test images are processed by the same classifier.

Using the setup described above and as an average of 3 independent runs for all categories, experiments were conducted in which at the beginning testing images were inserted as they were normally captured, treating them as “normal” at 0 degrees. Note that 0 degrees refers to the image at its originally captured position and not the dominant orientation. Subsequently, the images of this testing dataset are rotated at *45, 90, 135* and *180* degrees (Figure 5-28).

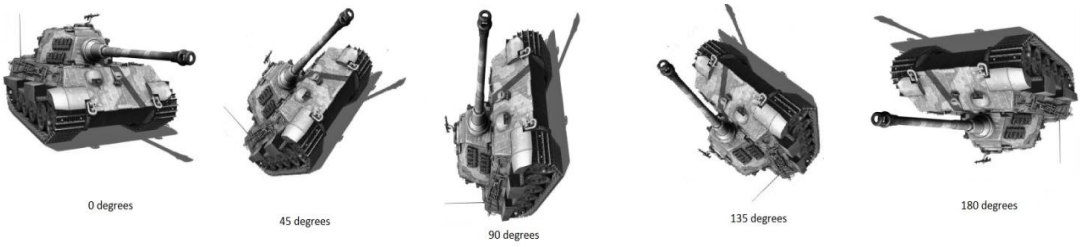


Figure 5-28: A test image is rotated and then used in the model for each of the rotation experiments in FHLib. From left to right, the tank is at 0, 45, 90, 135 and 180 degrees.

FHLib rotation results	0 deg.	45 deg.	90 deg.	135 deg.	180 deg.	Mean
Classification Accuracy (%)	87.5	26.5	23.5	28.5	56	44.4
Difference in (%) w.r.t. 0 deg	0	61	64	59	31.5	53.8
Percentage w.r.t 0 deg	100	30	26	32	64	38

Table 5-2: The average classification accuracies after 3 independent runs, under the CUUD dataset under different rotation angles using FHLib.

From Table 5-2, it can be deduced that as rotated images differ from the normal position at which the classifier was trained the results portray inconsistency and severe degradation in performance peaking at 90 degrees. On the other hand, the performance at 180 degrees stands out as the second highest among the rotation experiments mainly because of the symmetrical features that have been learned (i.e. wheels) or vertically and horizontally inverted features have not lost their spatial extent significantly. It is evident that the model is tuned to features which do not exhibit rotation invariance since classification accuracy should be ideally constant or minimal. Meanwhile, FHLib's limitation to explore maximum responses of features beyond a certain area from the original extraction area, illustrates that without a severe computational cost more consistent results cannot be achieved. There is no mechanism in the original FHLib which would compare features that differ only in rotation and which are otherwise identical. In real-world applications, it is fairly common for mobile cameras or even cameras at a fixed position to wobble due to redirection, reposition, handling and vibrations. In biological vision such real-world scenarios do not impact recognition performance significantly. A cortex-like computer vision model should follow this ability to a certain extent and keep with the object constancy abilities that humans possess.

5.4.3 Object orientation alignment in uncluttered environments

All experiments described in this section are a result of 3 independent runs, 50 training images per category (200 in total) and 50 testing images per category (200 in total) and of otherwise identical parameterisation to the previous section Following the theory presented in section 5.4.1.1, experiments are carried out using the rotation invariance feature (Table 5-3).

MFHLib rotation results	0 deg.	45 deg.	90 deg.	135 deg.	180 deg.	Mean
Classification Accuracy (%)	58	36	53	36.5	55.5	47.8
Difference in (%) w.r.t. 0 deg	0	22	5	21.5	2.5	12.5
Percentage w.r.t 0 deg	100	62	91	63	95	77.8

Table 5-3: The average classification accuracies after 3 independent runs, under the CUUD dataset using rotation invariance with MFHLib.

In Table 5-3, the results have been obtained by adding rotation invariance via object alignment and it can be seen that the maximum classification score at 0 degrees has dropped by almost 30% with respect to FHLib in Table 5-2. The reason behind this difference is that with object orientation assignment there is no “normal” position for the any of the images anymore. Instead, training and testing images rotate prior to entering the C1 layer of the model according to their dominant orientation and then according to equation (5-5) the object is rotated to 0^o. The only case where an image remains at 0^o is when its dominant orientation is already at zero. Furthermore, even though the general structure of the object has its orientation determined, localised orientations may vary even between objects of the same category. Since there is a random mixture of S2 feature sizes (i.e. 4x4...16x16) that have been learned at the training stage, these localised orientations of features will be random and undefined. As an alternative, making use of larger feature sizes, and comparing more general overall structures of objects, would in theory provide more accurate results.

For the rest of the rotation angles, there is a noticeable improvement in performance with respect to Table 5-2. The mean difference with respect to 0 degrees is significantly reduced compared to FHLib and at the same time, the overall percentage with respect to 0 degrees across all degrees remains at 77.8%, almost 40% higher than Table 5-2. This impressive improvement is acknowledged to come however, from only one uncluttered dataset. It is unlikely, that this behaviour could be maintained consistently in more complicated visual scenarios without some form of segmentation.

5.4.4 Local feature rotation invariance

Following the methodology presented in 5.4.1.2, the aim of the experiments in this subsection extends beyond the use of a single uncluttered dataset. The experimental setup is similar to 5.4.3 with a couple of differences. Firstly, due to the increased complexity introduced with multiple rotations and computations multiplied by an order of 12 (as the number of rotations from 0-360 degrees), the training images per class are reduced to 15 and the number of features per image to 10. This compromise was necessary due to the time constraints for

this particular topic. Secondly, the degree range spans from -135 to 180 in steps of 45 degrees.

Two versions of the algorithm are compared, MFHLib normal i.e. the direct and equally comparable implementation of the FHLib algorithm in Matlab without any modifications and MFHLib with Local Feature Rotation (LFR) modified as in section in 5.4.1.2. Four datasets in total were examined CUUD, CUCD, 10 class and CAL10. The dataset “CAL10” has the first 10 classes from the Caltech 101 (section 5.2.2).

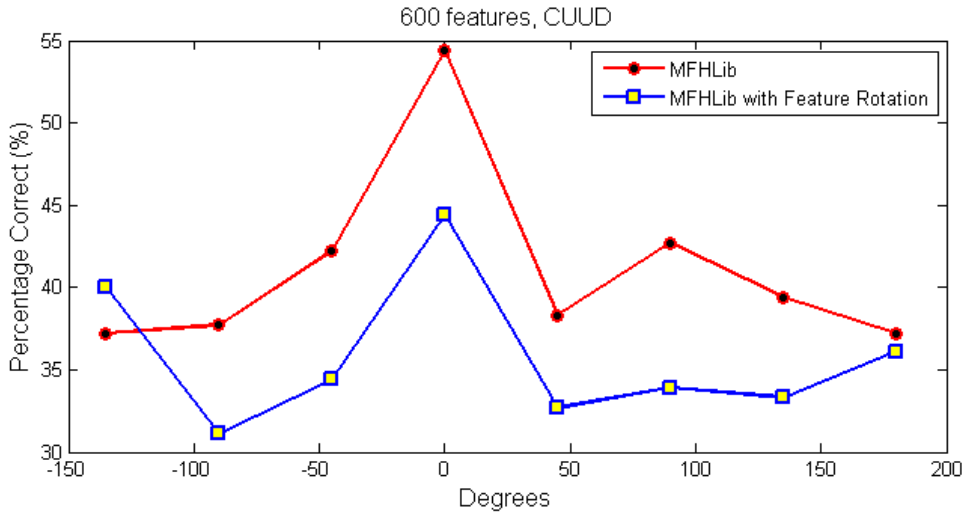


Figure 5-29: The average classification accuracies after 3 independent runs under the CUUD dataset, for MFHLib and MFHLib LFR. All results may typically vary at ±1.5%.

From Figure 5-29, it can be seen that the mean of MFHLib normal is approximately 41.1%, this is 5.4% higher compared to MFHLib LFR with a mean at 35.7%. The standard deviation for MFHLib rests at 5.7 while for MFHLib LFR a small improvement is noticeable at 4.3.

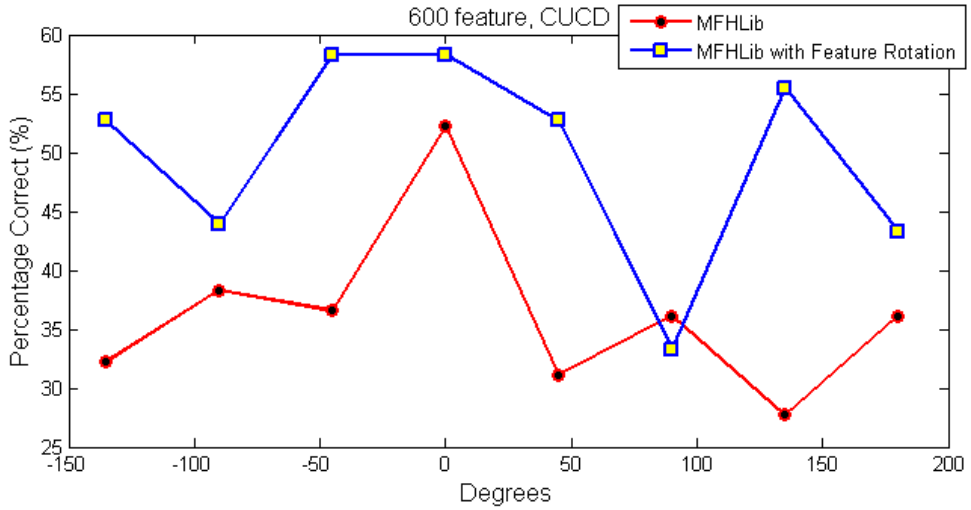


Figure 5-30: The average classification accuracies after 3 independent runs under the CUCD dataset, for MFHLib and MFHLib LFR. All results may typically vary at ±1.5%.

In Figure 5-30, a quite different set of results for MFHLib LFR is observed. The highest classification accuracy score is achieved at 0 degrees MFHLib LFR, with 58.3% as opposed to the 52.2% in MFHLib. Over the range of rotation angles, MFHLib LFR exhibits a more steady performance not only compared with MFHLib but also with results in Figure 5-29. This is evident from the average classification accuracy for MFHLib LFR in Figure 5-30 which is at 49.75% across all angles while it is at 36.3% for MFHLib. The mean value at 49.75% for MFHLib LFR with respect to the value at 0 degrees is significantly improved at only 8.5% lower. Nevertheless, there is a noticeable dip at 90 degrees which pushes the standard deviation to 8.8 as opposed to 7.3 in MFHLib.

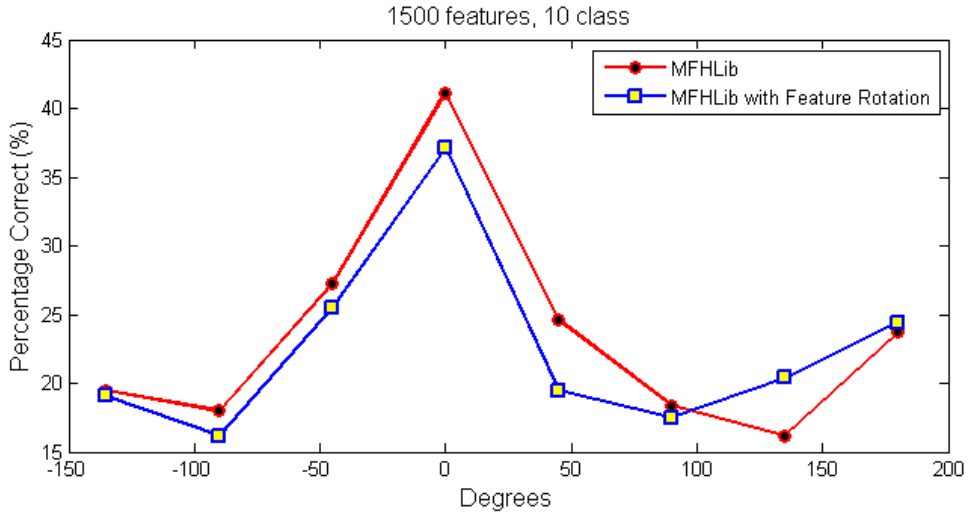


Figure 5-31: The average classification accuracies after 3 independent runs under the 10class dataset, for MFHLib and MFHLib LFR. All results may typically vary at ±1.5%.

The set of results between MFHLib and MFHLib LFR as in Figure 5-31 share a similar pattern, with MFHLib performing with a higher overall classification accuracy. The highest noticeable value is received at 0 degress MFHLib with 41.1% followed by MFHLib LFR with 37.1%. The similarity of results can be seen from the mean values for MFHLib and MFHLib LFR being at 23.6% and 22.5% respectively. Standard deviation values favour MFHLib LFR at 6.7 while for MFHLib it is slighty larger at 8.

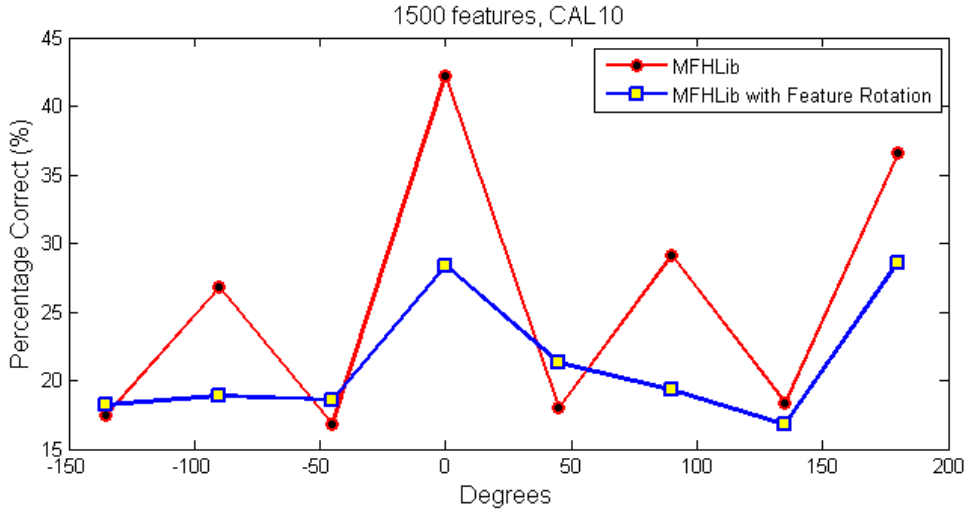


Figure 5-32: The average classification accuracies after 3 independent runs under the CAL10 dataset, for MFHLib and MFHLib LFR. All results may typically vary at ±1.5%.

In Figure 5-32 above, the highest classification accuracy of MFHLib for the CAL10 dataset has the greatest difference of 13.6% with respect to the

marginally highest classification accuracy of MFHLib LFR at 180 degrees. Similarly to the plotted results in Figure 5-31, the average value between the two algorithms is relatively close at 25.6% for MFHLib and 21.3% for MFHLib LFR. Standard deviation is lower the on other hand for MFHLib LFR with 4.6 as opposed to 9.7 for MFHLib.

In general, it is encouraging for the MFHLib LFR to exhibit the behaviour shown in Figure 5-30 for CUCD, but this has not been maintained across datasets. It seems that the choice of the total number of features or the features per image and the number of training images has not been sufficient to enhance performance in the bigger datasets. Even so, it remains ambiguous why CUUD in Figure 5-29 portrays a conspicuously different pattern to CUCD. Without an in depth experimentation with more computational power to incorporate more features, images and datasets, a safe conclusion for this mechanism would be premature.

5.4.5 Section Conclusions

Two different methods have been presented in this section. “Object alignment” attempts to find a dominant orientation from an object as a whole and the “local feature rotation” method rotates each feature at the S2 layer to find maximally responding values to each template in various rotation angles. A drawback for object alignment is that it requires either uncluttered backgrounds or an efficient segmentation mechanism, for more detailed experimentation. A major drawback of the MFHLib LFR has been its computationally expensive approach and lack of adaptation on different datasets. Any further decrease in the size of the rotational steps from 30 degrees and experiments with more features, images and datasets, would considerably increase computational time and stress.

Nevertheless, it is promising that object alignment, overall image rotation angles, has shown better performance compared to FHLib for the CUUD by nearly 40%. At the same time, for the more complicated cluttered background dataset CUCD MFHLib LFR showed a more consistent set of classification accuracies with respect to MFLib.

With more time and computational power, future experiments on this section’s algorithms could expand with more datasets, different parameterisations and scenarios. Further rotation methodologies should be also examined perhaps based on physiological data that provide more insight on the biological process.

6 RECOGNITION TASKS – SALIENT FEATURE-BASED AND COLOUR

6.1.1 Using GBVS with MFHLib

It was seen in section 5.3 that the combination of the two streams hypothesis has roughly divided the relative research work into two categories, segmentation-driven for detection and salient feature-based for object or scene recognition. The work in this section falls primarily in the second category and is a refinement of the basic FHLib model with particular attention to the main drawbacks inherited from the original architecture. Section 5.3 addressed the first step in the visual streams cooperation while the section here focuses on processes after the ROI of the image has been isolated and therefore subsequent stages of biological-like vision.

A major contribution of this section is the incorporation of saliency within the template feature extraction process and its implementation, using MFHLib as a foundation, is termed as Saliency FHLIB (SFHLIB). Other enhancements are:

1. The substitution of computationally expensive Gabor filters for multiple orientations with a single circular Gabor filter.

2. The improvement of the feature representation using larger patches and higher resolution.

3. The addition of an extra layer to refine the feature library thereby eliminating redundant patches while at the same time ranking features in the order of significance.

The classification in the present section has been achieved through a linear Support Vector Machine (SVM) classifier similar to the sections above. In section 6.1.2.2, the classification method is varied in order to validate the accuracy of the results under different schema.

6.1.1.1 Feature extraction from salient ROI

Unless there is a task in which even the most refined features are required to distinguish subtle differences or similarities between objects (often of the same category) then retaining all visual information is computationally expensive and unnecessary. In FHLIB and MFHLib (Appendix C) there is no specific pattern by which features are extracted and the selection process of both feature size and locations occurs randomly across input images. Moreover, it becomes difficult to estimate the required number of features or feature sizes. Solving this problem

by introducing a geometric memory in the algorithm (4.2.4) i.e. storing the location coordinates from which an *S2* template was found so that a respective area in a testing image is compared, led to the conclusion that such a system becomes specialised in recognising objects in similar locations [219]. This however is impractical for real-world situations since objects may appear at other locations or may become differently orientated and so the algorithm must generically overcome this problem.

By applying the GBVS model on a particular object, points to salient areas and evaluates an activation map according to priority of attention. For objects of the same category the most prominent areas are nearly the same and thus structured objects can be represented in a semantic way (Figure 6-1). In this section the orientation feature is only used and salient areas are ranked for a certain circumference (10 pixels) around the highest values. These areas effectively represent the local features which can be used along with global shape areas, i.e. larger types of features extracted freely from any point in the image and can be combined for the recognition stage.

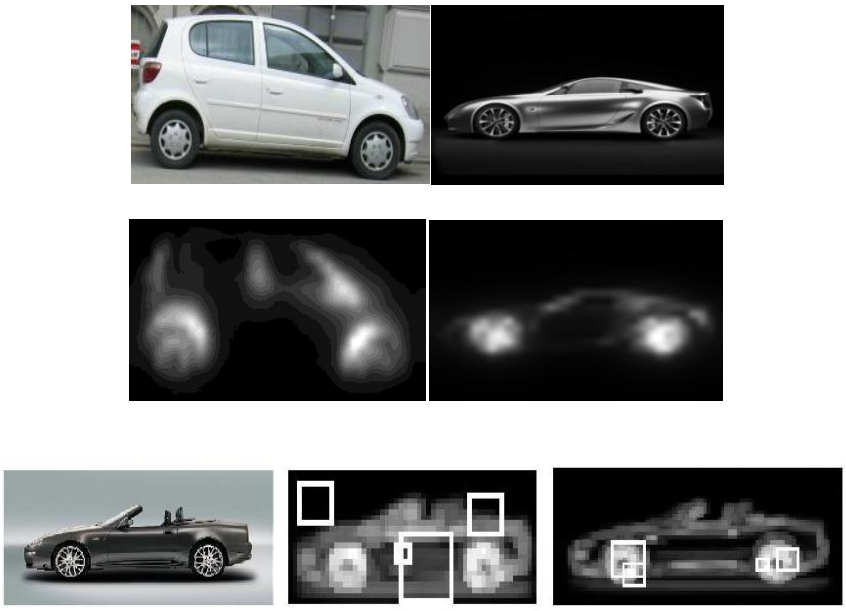


Figure 6-1: The top row shows the original images, the second row their saliency maps and the third row salient feature using GBVS and MFHLib in MATLAB.

In Figure 6-1, it can be seen that the highest attention accumulates on the wheel areas which is a common saliency feature and it is evident that saliency can effectively ignore background information. The third row shows the effect of accurate feature extraction via salience in a *C1* layer map. Rectangular boxes illustrate the feature templates of varying sizes. Extraction occurs in MFHLib

(centre) “blindly” while in a *C1* map from SFHLib (right) patches are extracted from the salient ROI.

6.1.1.2 Higher resolution, patch sizes and Circular Gabor filters

Salient areas can be very specific to small regions of an image. At low resolutions spatial information is also low and therefore extractions yield incoherent representations of the objects (Figure 6-2). To overcome this problem and to improve spatial representation, the resolution of images has to be increased. At the same time, in order to maintain a reciprocal spectrum of patch sizes by changing the image resolution, patch sizes are also increased. To tackle these issues, the size of the short edge of an image was increased to 240 pixels (thus preserving the aspect ratio) and included *S2* feature patches of sizes 20x20, 24x24 and 28x28.

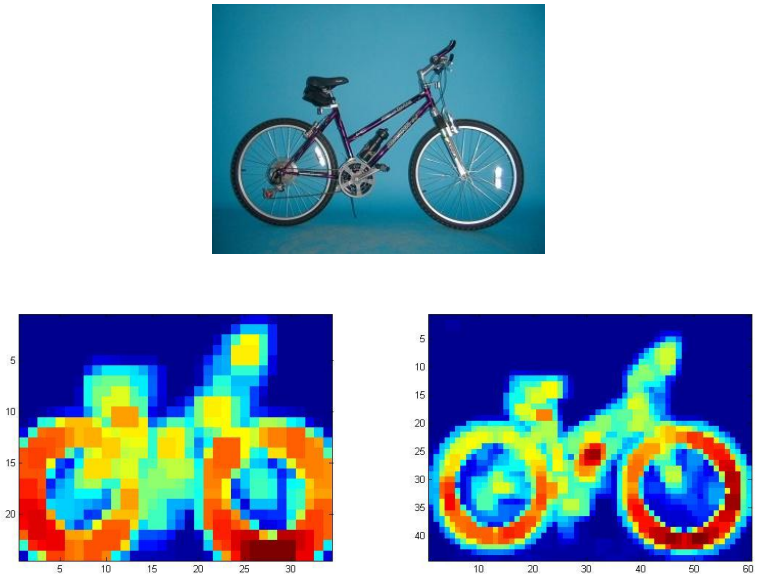


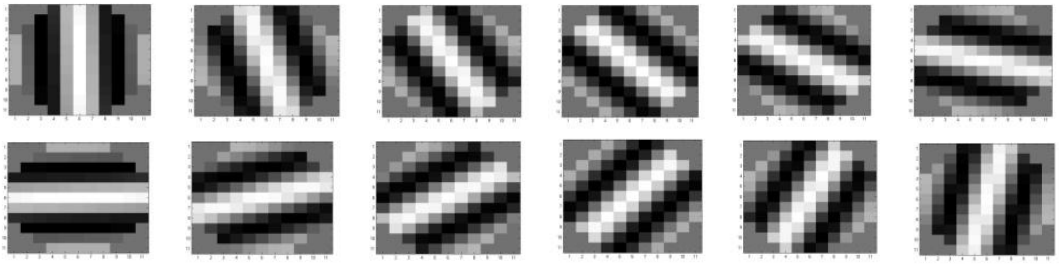
Figure 6-2: An example of an original image top, at a lower resolution the C1 layer (left bottom) has retained little of the object’s structure while at a higher resolution spatial clarity at the C1 layer is apparent (right bottom).

The use of Gabor filter banks in object recognition adequately simulates the tuning of V1 simple cell at different orientations *(θ)* and highlights their role in bottom-up mechanisms of the brain (Section 3.4.2). However, constructing *S1* responses for different orientations requires the creation of an equal amount of Gabor pyramids for each orientation which is computationally expensive and time consuming as the number of orientations increases to improve an object’s description. To eliminate this, the *S1* responses are generalised by varying the sinusoid across all orientations which then becomes circularly symmetric [220]. Using this single circular Gabor filter, one S1 pyramid is obtained and at the

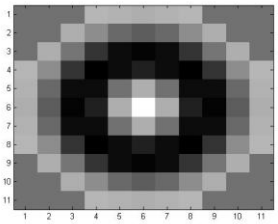
same time FHLib's sparsifying step over orientations (section 4.2.4) becomes redundant and is removed. The circular Gabor filter is given below:

$$G(x,y) = \exp\left(-\frac{X^2+Y^2}{2\sigma^2}\right)\cos\left(\frac{2\pi}{\lambda}\sqrt{X^2+Y^2}\right) \qquad (6\text{-}1)$$

Note that in equation (6-1), θ and γ are no longer parameters for the Gabor equation and the equation now only depends on σ and λ (the values for σ and λ, are the same as in [189]). Furthermore, in equation (6-1), X is simply the value of the radius along the x direction and Y along the y direction. Figure 6-3 below, shows the circular 11x11 pixel Gabor filter template versus the 12 orientation standard Gabor filters previously used.



(a)



(b)

Figure 6-3: (a) Gabor filters at 12 orientations (b) One circular Gabor filter. γ, σ and λ for both methods are set according to [189]**.**

6.1.1.3 Adding S3 and C3 layers

At the object recognition part of the model, when training template patches are extracted randomly from salient ROI, it is inevitable that patches are extracted more than once from the same location and scale, especially as the required total number of training patches is increased. Furthermore, there is no refinement mechanism in MFHLib that evaluates the extracted patches' performance and as such the algorithm may store patches that do not explicitly and accurately represent each class. In the original FHLib algorithm, a

refinement was made at the classification stage [189], however it is a post-processing, non-biological and time consuming remedy.
Both aforementioned issues are addressed with the introduction of two more layers, *S3* and *C3*. In the *S3* layer, all patches of a particular class are directly grouped together from the *S2* featurebook (section 4.2.4) and are organised according to their extraction sequence. The algorithm continues by extracting the training *C2* vectors (as in MFHLib) which are again grouped so that the responses of every patch from each class across all images now exist together. By examining the *C2* responses of each patch for every class on objects of the same class, e.g. if the class was 'bikes' and a patch was extracted from one of its images then *C2* responses for this patch from all images portraying bikes are grouped together. Patches that have yielded identical *C2* responses (in practice *C2* vectors are float numbers and identical responses can only be obtained from identical patches) are dropped and only one unique patch is retained therefore eliminating co-occurrences. The origin of the retained *C2* vectors is stored and refinement of the *S3* featurebook is done accordingly.

Additionally, the performance of each patch can be measured for every class against objects of the same category to deduce the sampled patches that best describe that class. By summing the *C2* responses for every patch, *S3* patches are ranked from high to low (high showing patches that are most commonly found for a particular object, low showing less occurrence and so uncommon patches that do not exist across all images). At this point, a percentage number is introduced, an amount of patches to be retained and for example, a certain value means that the featurebook is reduced by that percentage and the patches retained maximally express the trained classes. The final version of the significantly reduced *S3* featurebook refined from co-occurences and uncommon patches is used over the training images to create *C3* vectors which in turn are used to train the SVM classifier.

To validate the recognition performance with the extra layers *S3* and *C3* introduced here, a series of experiments were conducted in order to examine the effect of the feature reduction percentage. In Table 6-1, four datasets are used, "CUUD", "CUCD", "10 class" and "CAL10". Each row in Table 6-1 shows a step decrease of 20% in the total number of features retained. The experiment was set up for 15 images per category for both training and testing, with 50 features per image as a prerequisite, which leads to a variable total number of features for each dataset. Classification was carried out with an SVM using one-against one decomposition (*gamma* =1, *C* = Inf) as in the previous chapter.

Dataset / Method	CUUD (initial 3000 features)	CUCD (initial 3000 features)	10class (initial 7500 features)	CAL10 (initial 7500 features)
SFHLIB -100	62.22 - (2380, 2292, 2396)	62.8 - (2660, 2676, 2680)	47.3 – (5900, 5900, 5730)	42.4 – (5140, 6740, 6380)
SFHLIB - 80	58.9 - (1856, 1880, 1872)	58.9 - (2152, 2172, 2140)	47.5 – (6030, 4670, 4640)	40.9 - (3930, 5240, 5270)
SFHLIB - 60	60 – (1400, 1384, 1400)	58.3 - (1608, 1620, 1596)	46.7 – (3430, 3430, 3490)	37.8 - (2640, 3940, 4060)
SFHLIB - 40	59.4 – (952, 932, 912)	60 - (1080, 1056, 1080)	46.4 – (2270, 2270, 2370)	40.2 - (2590, 2590, 2590)

Table 6-1: The average percentage of classification accuracies over 3 independent runs while varying the feature reduction percentage downwards from 100 in steps of 20. In brackets, the remaining feature numbers for each run.

Across all datasets from Table 6-1, it is seen that even after the significant reduction of the featurebook, the lowest percentage SFHLib 40 produces comparable and consistent results compared with SFHLib 100. The featurebook reduction by 60% for example in CUUD shows only a 2.8% decrease in performance. Another element to notice is the difference of feature numbers between CUUD and CUCD, since CUUD is an uncluttered dataset it follows that the amount of duplicates will be higher.

Figure 6-4 to Figure 6-7, illustrate the results of using SFHLib 40 under a different number of training images per class, with the same number of features. In all figures, the consistent rising trend of classification accuracies as the number of images increases is apparent. In essence, these figures show that for a smaller and more computationally-economic percentage of semantic salient features with the use of the extra layers *S3/C3*, a higher percentage of classification accuracies can be achieved by increasing the number of training images per class. In Figure 6-7, experiments for CAL10 were only conducted up to 30 images per category because the total number of images was insufficient.

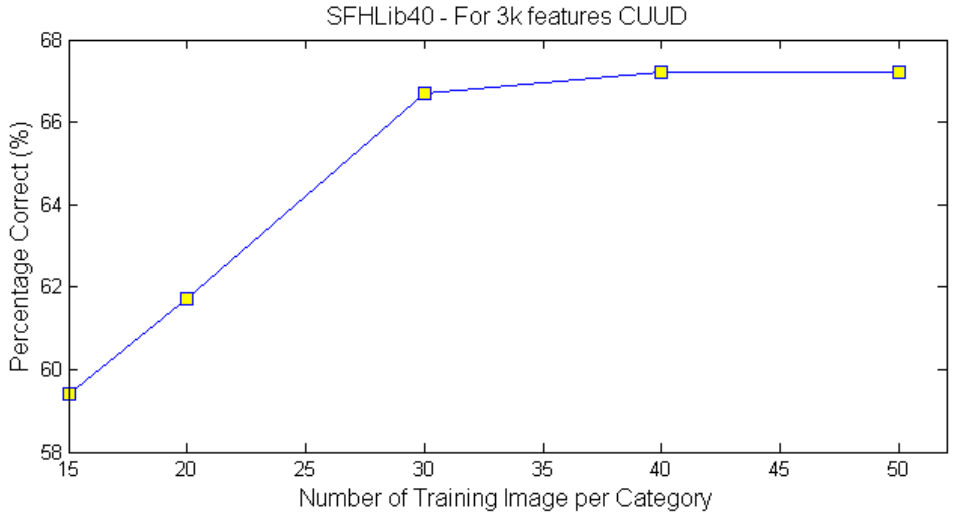


Figure 6-4: The average classification accuracies after 3 independent runs, for SFHLib 40 using the CUUD dataset at various numbers of images per class.

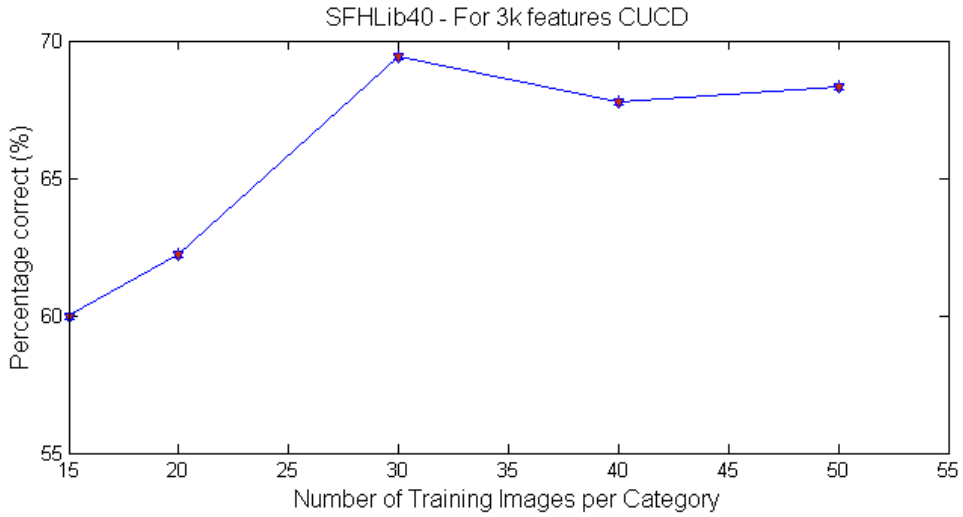


Figure 6-5: The average classification accuracies after 3 independent runs, for SFHLib 40 using the CUCD dataset at various numbers of images per class.

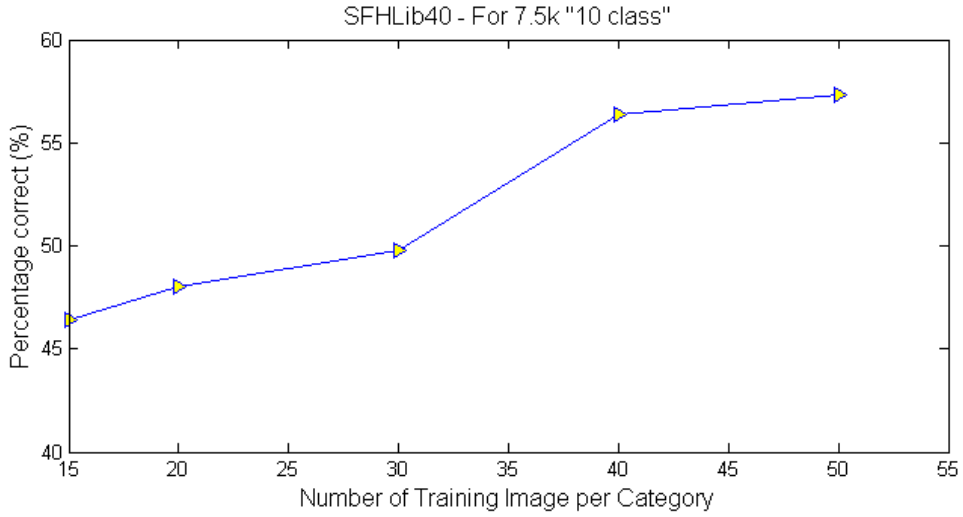


Figure 6-6: The average classification accuracies after 3 independent runs, for SFHLib 40 using the 10 class dataset at various numbers of images per class.

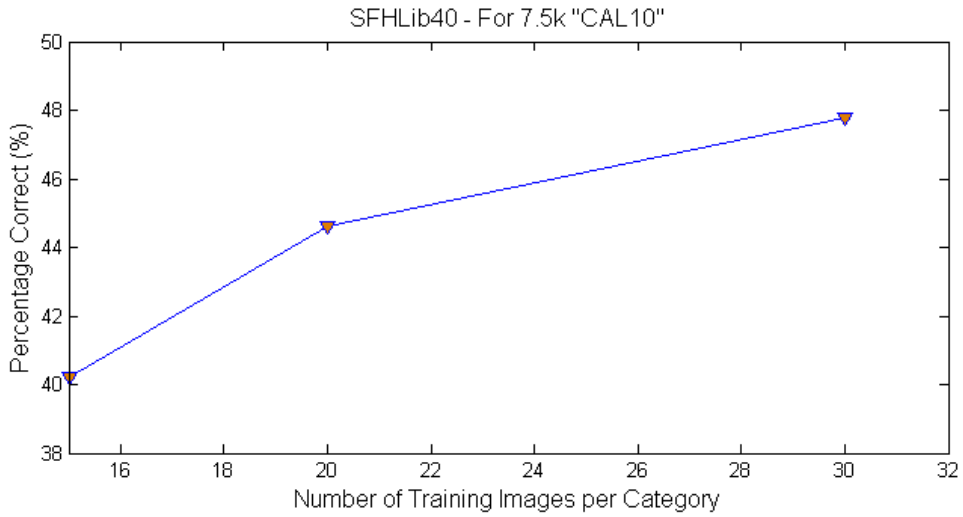


Figure 6-7: The average classification accuracies after 3 independent runs, for SFHLib 40 using the CAL10 dataset at various numbers of images per class.

6.1.1.4 SFHLib Experiments Setup

Three image datasets are used CUUD, CUCD and Caltech 101. GBVS MATLAB code and default parameterisation (Appendix B) is directly used with some modifications while all code regarding the recognition part relies on MFHLib (Appendix C).

The algorithm is first tested with FHLib-like parameterisation, 140 pixels for the images' short edge and 4 different size patches (4x4, 8x8, 12x12, 16x16),

11x11 Gabor filter banks while a sliding window approach was used to extract the maximum *C2* responses across the entire image. At this point, the Gabor filters consist of 12 banks i.e. one per orientation angle. Subsequently, the algorithm is enhanced gradually by introducing a higher resolution for each image (240 pixels, short edge) and adding three more patch sizes 20x20, 24x24 and 28x28. The salient feature extraction method is then introduced and the 12 Gabor filters are substituted with one circular Gabor filter. Finally, the *S3* and *C3* layers are in turn embedded and tested.

Efficient and fast biological-like detection and object recognition requires parallel execution. Experiments here are result-driven and therefore a sequential approach is used. Hence, the saliency maps of both training and testing images of all datasets are prepared beforehand.

Each saliency map from GBVS exactly matches the size of the original image later used in object recognition i.e. 240 pixels for the shortest edge, and the only feature used is orientation at 12 Gabor angles spanning from 0 to π. For all experiments during training, an abundant number of features was chosen (10000) to avoid underrepresentation of objects and Gabor filter parameters γ, σ and λ are all fixed according to [189]. For datasets CUUD and CUCD, 50 different images for each class were chosen for training and another 50 per class for training. For the Caltech dataset, 15 images per class were chosen for training and 15 per class for testing. Classification accuracies are obtained as an average of 3 independent runs for all experiments.

Method \ Dataset	CUUD	CUCD	Caltech
MFHLIB	80	70.6	18.75
SFHLIB + Circular Gabor	85	80.4	22.4
SFHLIB + S3/C3 Layers (60% features)	86	76.6	19
SFHLIB + S3/C3 Layers (100% features)	81	80.4	21.4

Table 6-2: Average percentage of classification accuracies over 3 independent runs for the three datasets. Note that descending order algorithms in the left column include the enhancements of the previous algorithms. All results typically vary at ±1.5%.

In Table 6-2, the results portray for all enhancements a gradual improvement over both the accuracy itself and time. CUUD being uncluttered, presents minimal difficulty for an algorithm and classification accuracies were overall the highest. Under this dataset, a 6% percentage improvement was observed between MFHLIB and SFHLIB variants (excluding SFHLIB with 100% features).

A higher difference between the MFHLIB and the enhancements here was noticed in CUCD. In this dataset even though the number of classes remains the same, the added background information and more complicated poses, affect the performance of all algorithms, particularly MFHLIB. As a first step by increasing the resolution and changing the number and size of patches has increased performance by 6% and a total of 10% better performance was achieved by using SFHLIB with circular Gabor filters. A drop of nearly 10% for MFHLIB between CUUD and CUCD signifies its inefficiency as a dataset becomes more realistic. A decrease in performance (4.5%) can be also observed for SFHLIB though it is almost half compared with MFHLIB.

Experiments with the benchmark Caltech 101 dataset have revealed a decrease in performance with respect to the other two datasets which was primarily caused by the large number of classes and different setup. However, within this set of experiments an incremental difference between FHLIB and SFHLIB is apparent.

Classification accuracies for *S3/C3* layers show that although for the cluttered datasets an improvement can be claimed the trend is not followed in CUUD. A major difference between previous variants of the code is that the number of features required to achieve this performance was lower and thus computationally cheaper. Having selected a fixed number of features (10000) for the library, by running the *S3/C3* on the CUUD, reductions of an average of 15% were observed for a 100% of the features used. Similarly for the CUCD, the average percentage of identical feature discards reached 22% and for the Caltech dataset 10%. The difference of this percentage between the three datasets can be explained by the larger number of images used in the Caltech data. The same total number of features corresponds to fewer features per image thus reducing the probability of identical patches extracted randomly across salient regions. Discarding identical features improves time (by approximately the same percentage) and computational requirements.

6.1.2 Improving salient feature based object recognition

Following the experimental conclusions of the previous section, certain additional steps were taken to improve the performance of the algorithm and the accuracy of results, so that:

(a) The visual attention model is refined and incorporated as part of the algorithm.

(b) Spatial distribution of salient features is retained.

(c) Experiments are enriched with more datasets.

(d) Feature reduction analysis has been added.

(e) Improvements are validated under different classification methods.

6.1.2.1 Cranfield University Visual Salience (CUVS)

In this set of experiments, GBVS is completely removed from the algorithm. Inspired from IKN and GBVS, the saliency of intrinsic features is obtained without any intentional influence while incorporating a very important characteristic of biological vision, the centre-surround operation. Initially for a digital RGB input image, the new model extracts fundamental features i.e. average intensity over the RGB bands, double-opponency colour and Gabor filter orientations. Conspicuity maps are formed from using image pyramids across different scales under centre-surround operations. As in the previous section, the orientation feature is considered:

$$O(c,s) = \left| O(c) \Theta O(s) \right| \tag{6-2}$$

$$\overline{O} = \sum N\left(\bigoplus_{1}^{7} N(O(c,s)) \right) \tag{6-3}$$

Equation (6-2) expresses the across scale differences of the Gabor-filtered pyramid, *c* (centre) is for scales 1-3 and *s* (surround) is for scales 4-10. Resulting maps are then, in equation (6-3), normalised (shown with the symbol *N*) and across scale added. This procedure is illustrated in Figure 6-8, the RGB image is converted into an intensity representation and after using a circular Gabor filter, the algorithm applies the procedure above. After normalisation and across scale addition, a slight Gaussian blur is applied in order to broaden the saliency areas and to merge the closely neighbouring ones. The final saliency map highlights the most prominent ROI in the image and is used as a topographic map of salient feature information which is used later in recognition.

The process of converting the image to intensity, using Gabor filters and creating pyramids is shared with the recognition part of the algorithm. Unlike past studies in which salience would feed recognition sequentially with no common processes and interaction, in this section, salience and recognition

share common processes up to the formation of Gabor pyramids. This follows from knowledge on the biological process (chapter 4).

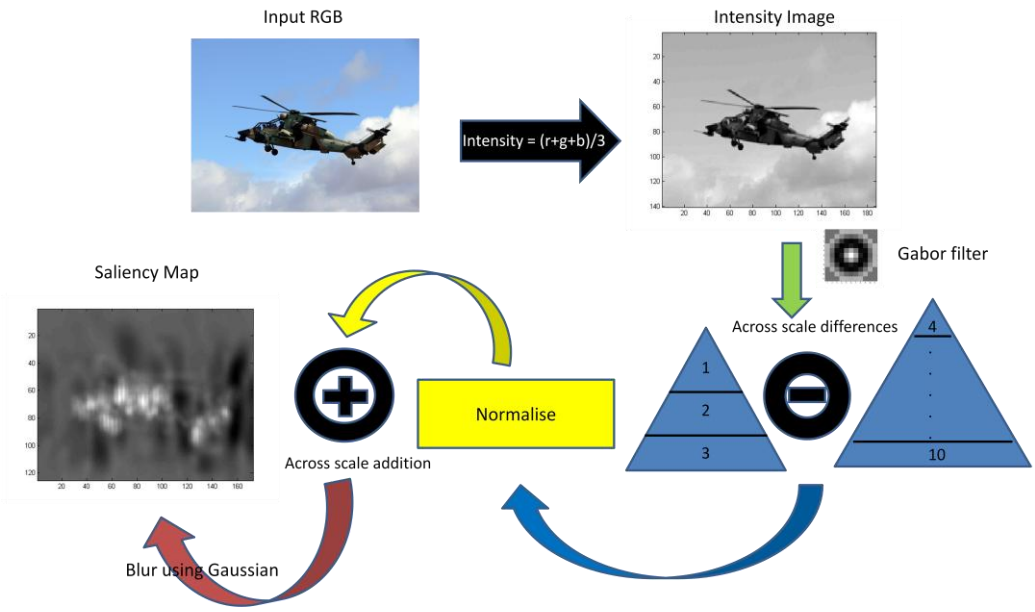


Figure 6-8: The Saliency map extraction technique.

6.1.2.2 Experiments on improved SFHLib

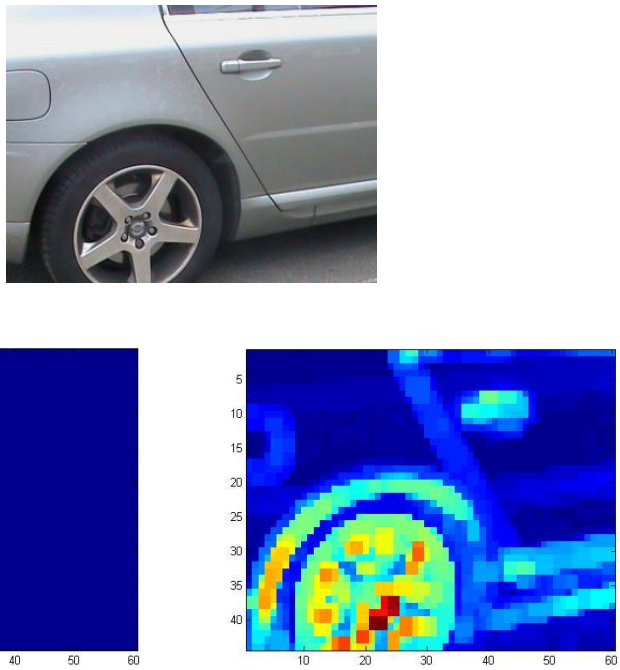
The number of image datasets is increased to four, and the Caltech dataset is removed completely for having too many classes and too few examples in each for the particular experiments here. More specifically, the chosen datasets are:

- CUUD
- CUCD
- 10 class dataset which includes the classes from CUCD.
- 25 class dataset which in turn includes the classes of the 10 class dataset and CUCD.

For all datasets, 30 different images for each class were chosen for training and a different set of 30 images per class for testing. This means that although in each dataset the total number of features in the featurebook is varied (i.e. 6000 features for the 4 class datasets, 15000 for the 10 class and 37500 for the 25 class), the number of features per image is always 50. This allows the measurement of the performance to be free from spatial under-representation or irregularities brought in by various numbers of features per class.

MFHLib and the algorithm used here in its primary form maintained a threshold-like operation that inhibited *C1* units. Specifically, as in equation (4-44), at each location the minimum and maximum values (R_{min} and R_{max}) are used so that if any unit is below a threshold value, it is suppressed to zero (0.5 default value).

The effect of this inhibition is illustrated in Figure 6-9. This threshold function can, depending on the image and dataset, provide some background elimination but on the downside may also remove necessary visual information from the scene. It was created under the assumption that for a simplistic dataset like Caltech 101, the image would portray a very distinct object around the centre of the image, making background suppression easier and intuitive. However, when the visual scene contains more than one object of interest or when these objects are scattered around then background thresholding would eliminate them completely or distort them. It was found that there was a conflict when using salience with this mechanism, since often salient features might be introduced from what the inhibition mechanism considered as “background” and therefore suppressed to zero. Consequently, the inhibition value h is set to 0 in order to override the process and the hypothesis made here is that the inhibition mechanism should be used differently to be useful in more complex datasets.



(a) (b)

Figure 6-9. (a) Shows *C1* map with inhibition constant at 0.5 (default), (b) Shows *C1* map with inhibition constant at 0 (no thresholding), spatial richness and integrity of (b) over (a) is clear.

At the *S2* layer for both training and testing, the radial basis function’s sigma used to perform template matching was set from 1 to 0.1. This was done in order to accommodate distinctiveness in responses that are very close together. Finally, classification accuracies are obtained as an average of 3 independent

runs for all experiments. The procedure is summarised below and in Figure 6-10:

1. Train path (black route in Figure 6-10) receives the input train images.
2. Apply edge detection with the circular Gabor filter.
3. Perform salience according to section 3.1.
4. Isolate salient ROI within the object then SFHLib extracts S2 templates from these areas following the procedure explained in section 2.
5. Max responses of S2 templates produce C2 vectors which in turn train the SVM classifier.
6. In the testing stage (red route in Figure 6-10) steps 2 and 3 are identical.
7. Salient ROI coordinates within the object are retained. Template matching is performed over these areas only.
8. C2 testing responses are fed into the same SVM classifier to produce recognition result.

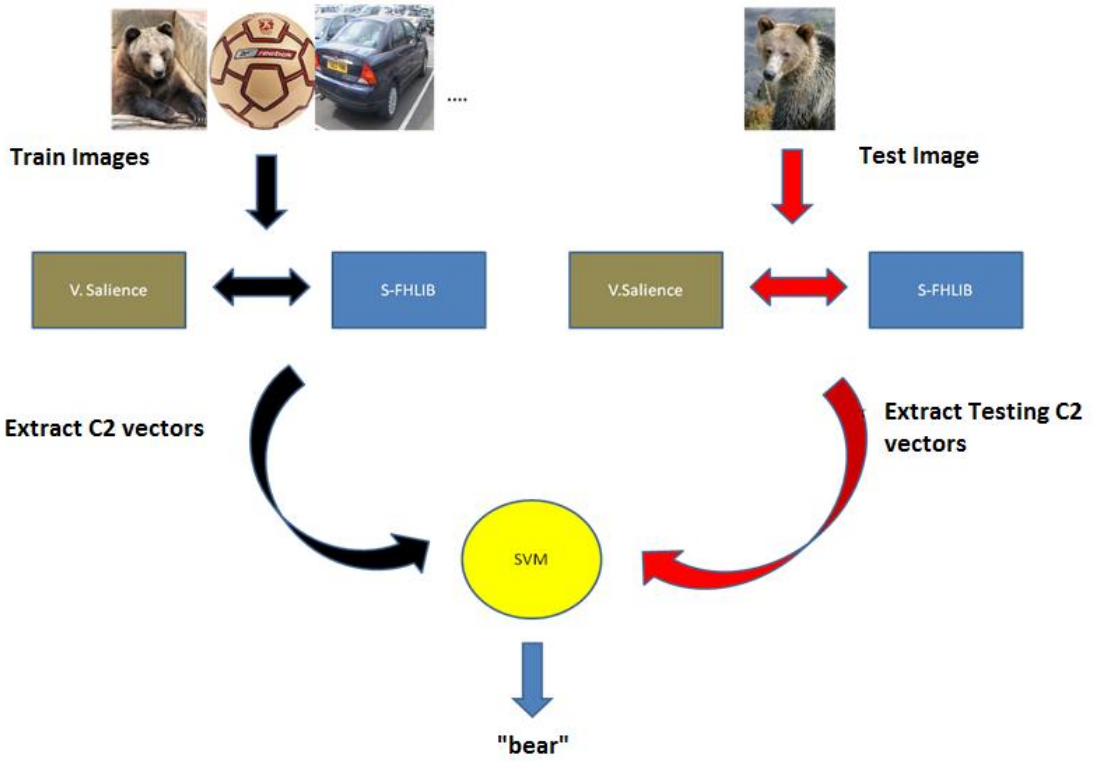


Figure 6-10: The general layout of the algorithm.

6.1.2.3 Results – Discussion

MFHLib is the foundation upon which all improvements of this work were established. Building on MFHLib gradually, further enhancements that were proposed in the previous section were added in the experiments (Table 6-3 and Table 6-4). The difference between Tables Table 6-3 and Table 6-4 is that the former contains the list of experiments that were conducted under inhibition 0 (no thresholding), while the latter provides a sample of experiments under 0.5 inhibition (default – i.e. background suppression) for comparison. From Table 6-3, it can be seen that the highest percentages in classification as expected were achieved from the four class datasets. This follows intuition since a four class dataset is trivialised compared to other larger multiclass datasets and should not provide substantial difficulty. It is also noticed that as the number of classes increases the classification accuracy percentage falls. Even though therefore, the number of features per image for each dataset remains the same it is obvious that introducing more classes prohibits similar performance amongst datasets. Ideally, regardless of the number of classes this line should have similar percentages.

Dataset / **Method**	CUUD	CUCD	10 class	25 class	Mean over datasets (%)
MFHLib	72	64	38	23	49.2
MFHLib + Circular Gabor, Larger Patches, Higher Resolution	79	68	52	35.2	58.7
SFHLib + Circular Gabor Larger Patches, Higher Resolution	79	67	54	36	59.2
SFHLib + S3/C3 Layers (100% features)	82.2	66	53	36	59.3
SFHLib + S3/C3 Layers (80% features)	80.6	67	51	35.2	58.5

Table 6-3. Average percentage classification accuracies over 3 independent runs for the four datasets with inhibition at 0. Note that descending order algorithms in the left column include the enhancements of the previous algorithms. All results typically vary at ±1.5%).

Dataset / Method	CUUD	CUCD	10 class	25 class	Mean over datasets (%)
MFHLib	85.5	70	39	28	56.2
MFHLib + Circular Gabor, Larger Patches, Higher Resolution	74	64	41.5	23	50.6

Table 6-4. Average percentage classification accuracies over 3 independent runs for the four datasets with inhibition at 0.5. Note that descending order algorithms in the left column include the enhancements of the previous algorithms. All results typically vary at ±1.5%.

In the second row in Table 6-3, certain elements of MFHLib are maintained (i.e. 140 pixels for the short edge of the image, randomised feature extraction) while the circular Gabor filter, higher resolution and larger variety of patches is introduced. In this row, the percentage increases substantially which shows the inadequacy of using smaller *C1* maps and a small variety of patches. It should be mentioned here that from these experiments it was observed that the substitution to a circular Gabor filter not only aids classification scores but also computational speed since rather than computing 12 orientation pyramids only one is necessary and as such, it shows superior performance over the multiple orientation filter banks. Multiple filter banks symbolise real V1 simple cell activity more closely but with the circular Gabor a more symbolical behaviour is achieved. The highest percentage difference was observed in the 10 class with an average of 14% better after the increase of resolution and patches, closely followed by the 12% noticed in the 25 class dataset. A smaller but nevertheless significant improvement was noticed for the 2 four class datasets as well. Comparing the first and second row from Table 6-3 with Table 6-4, certain differences can be evident. MFHLib shows superior performance (7% on average better) with respect to Table 6-3. After thresholding substantial spatial information has been removed (Figure 6-9) and any change on image size and patches has no improvement over the results. In fact, in the second row of Table 6-4 there is a noticeable decrease in performance. The mean over datasets as seen in Table 6-3 for the second row is at 8% higher than in Table 6-4. It is because of the theory in section 4.2 and evidence from Table 6-4 that for the remainder of the experiments inhibition is at 0.

With the introduction of salience, a slight increase in the algorithm's behaviour was noticed which is consistent with the previous row of experiments. The achievement here is because the algorithm no longer retains a random feature extraction method. This idea paves the way for the next two rows of

experiments. The goal is primarily twofold, on the one hand, to provide a higher mean over the datasets and consistency between multiclass datasets but also on the other hand, to minimise the size of the featurebook as much as possible therefore reducing computation time and memory efficiency without sacrificing from the overall performance. So, if the representation of objects through salience offers the desired efficiency as seen in the third row of Table 6-3, then it is logical to pursue the next steps of experiments with the introduction of *S3/C3* layers. With a percentage of 100% in *S3/C3*, the algorithm discards identical features from co-occurrences only and ranks them regardless, in case the performance of each feature needs to be seen or the percentage is changed. The reduction in number of features is shown in Table 6-5 below.

Dataset / **Method**	CUUD – Total number of features value 6000	CUCD– Total number of features value 6000	10 class– Total number of features value 15000	25 class– Total number of features value 37500
SFHLib with 100%	4720 ~ 21.3%	5420 ~ 9.6%	12070 ~ 19.5%	30100 ~ 20%
SFHLib with 80%	3770 ~ 37%	4288 ~ 28.5%	9925 ~ 33.8%	24200 ~ 35.4%

Table 6-5. Final feature numbers at S3/C3 layers over 3 independent runs for the four datasets with inhibition at 0. Percentages show the amount of reduction from the initial total number of features. Remember that datasets' feature values vary in order to preserve the 50 features per image criterion of these experiments.

From Table 6-5, it can be seen, for example, that for SFHLib with 100% of features after discarding identical features, the reduction of the featurebook's size is 21% from 6000 to approximately 4720 features, and that the library integrity is thus improved since the classification accuracies remain on average the same for the 100% case as can be seen in Table 6-3. Percentagewise the reduction of features for the rest of the datasets is smaller, since they contain background clutter to a greater extent. Arguably, the amount of feature reduction illustrates the amount of variance and richness that exists in a given dataset and may be seen as one criterion to judge the overall dataset integrity or the amount of features required for accurate representation. In other words, low feature reduction percentages mean a greater number of unique salient features being extracted and/or pose and object variability.

SFHLib at 80% feature reduction means that after the algorithm has discarded identical features, it is set to retain the top 80% based on their overall performance, i.e. the 80% of the maximally responding features. In this case, the last row of Table 6-3 shows that the reduction of performance is marginal even though as seen in Table 6-5, the featurebook has been reduced

significantly with the highest being at CUUD 37% closely followed by the 25 class' 35.5%. Any unregulated and random discard of features to the same percentages would have detrimental effects if these features are vital elements for the object's spatial integrity.

So far the classification process was examined under a SVM with a linear kernel. In the next round of experiments SVM with two other popular kernels is used, the radial basis function (RBF also known as Gaussian) and sigmoid, both presented in Table 6-6 below. Using a cross-validation technique based on accuracy the best values of *C* (regularisation constant) and *gamma* for each dataset are obtained (Table 6-7). "Gentle" Adaboost is also used to illustrate the results under a different classifier. In Adaboost, 100 weaklearners (simple perceptrons) at 1000 iterations are used, a number sufficiently high to ensure training accuracy, in the sigmoid function $\varepsilon = 1$ and regularisation parameter for the weight update $\lambda = 0.001$.

From Table 6-6, the trend observed under a linear SVM is preserved throughout the table. In some cases, this pattern is exaggerated, for example in CUCD under a linear kernel the maximum difference between MFHLib and the improved SFHLib is relatively small (3%) compared to the behaviour seen from Adaboost (30%) for the same dataset. In the 10 class dataset the noticeable improvement of approximately 15% between MFHLib and SFHLib is reduced in other classification approaches, e.g. with the RBF kernel all three methodologies are nearly the same. SVM linear and SVM RBF show the best performance percentagewise. Importantly, in all classifiers SFHLib at 80% shows consistent and comparable performance with respect to SFHlib with all the features present. It is arguable whether a one-for-all percentage is the best strategy for redundant or unimportant features in all classes. In the future, an experiment in which feature reduction is adapted for each class individually might reveal more efficient mechanisms. Nevertheless, results here have shown that by learning salient features in a ranked approach better results can be produced with respect to the random approach in FHLib and further experimentation with additional enhancements is necessary.

Dataset / Method	CUUD (%)	CUCD (%)	10 class (%)	Mean over datasets (%)
MFHLib - SVM Linear	72	64	38	58
SFHLib 100% - SVM Linear	82.2	66	53	67
SFHLib 80% - SVM Linear	80.6	67	51	66.2
MFHLib - SVM RBF	61	69.3	45.5	58.6
SFHLib 100% - SVM RBF	76.9	74.7	45.8	65.8
SFHLib 80% - SVM RBF	77.9	76.8	47	67.2
MFHLib - SVM Sigmoid	65.8	66.4	42.4	58.2
SFHLib 100% - SVM Sigmoid	76.3	66.1	45.4	62.6
SFHLib 80% - SVM Sigmoid	73.4	69.3	45.5	62.7
MFHLib - Adaboost	58.8	42.7	43	48.1
SFHLib 100% - Adaboost	68.8	69.3	46.8	61.6
SFHLib 80% - Adaboost	69.3	72	47.5	62.9

Table 6-6. Classification accuracies from the final version of the SFHLib method under different classification techniques as an average over 3 independent runs on CUUD, CUCD and 10class datasets. All results typically vary at ±1.5%.

Dataset / Kernel	CUUD	CUCD	10 class
Linear			
C	∞	∞	∞
gamma	1	1	1
RBF			
C	1000	100	100
gamma	Def.	Def.	Def.
Sigmoid			
C	100	1	1
gamma	Def.	Def.	Def.

Table 6-7. The best values of C and gamma for each kernel as found via cross-validation for each dataset separately. (Def. = 1/number of features)

6.1.3 Section conclusions

One significant contribution here has been to improve the object recognition performance by incorporating visual saliency into the ventral stream process. In this preliminary study on SFHLib, an enhanced fusion of salience and recognition, improvements of ~8% classification accuracy for the CUUD, 5% for the CUCD, 14% for the 10 class dataset and 12% for the 25 class dataset with respect to MFHLib were noticed when using a linear SVM. It has also been proven that this behaviour is consistent when employing other SVM kernel functions and classifiers. Directly comparing the different classification methods showed that overall, the linear SVM classifier performs comparably with an RBF kernel SVM but can be outperformed from the other classification methods depending on the dataset.

Another important aspect of this work has been the addition of extra layers into the main recognition algorithm, which eliminated repeated and insignificant or "noisy" salient features in order to purify the structural form of the objects. With the introduction of the two additional layers in the recognition part of the algorithm, the present work has highlighted the need of an efficient and hierarchical feature extraction method. Further alterations on the mechanism of the algorithm have revealed the significance of refining extracted features and ranking them for enhancing the integrity of the feature library without sacrificing in performance. It is planned to experiment in the future on an adaptive class-by-class feature reduction approach and compare it against the generalised method here.

With just a single Gabor filter and a smaller library of features it has been found that computational time and memory requirements of the proposed SFHLib have improved by a significant factor compared to MFHLib.

6.2 Colour in cortex-like object recognition

6.2.1 Cranfield University Visual Salience (CUVS)

In the original FHLib model and consequently on the early version of MFHLib, target recognition relies only on morphology features. This information is randomly extracted across the *S2* layer in the form of templates as seen in the previous section. This random method was treated as undesirable in this thesis and the extraction of features was instead executed from salient ROI. This methodology was termed in the previous section as Salient FHLib or SFHLib. In contrast to previous work in [189] where only morphology feature maps were used in saliency (colour features would have been unused due to the construction of MFHLib and SFHLib), colour feature extraction is introduced here in both salience and recognition. An object's intrinsic features are obtained via saliency without any intentional influence while incorporating two important characteristics of biological vision, double-opponency and centre-surround operations.

The procedure followed here is also outlined in Appendix C. Initially for a digital RGB input image, the algorithm extracts the average intensity as in equation (6-4) which is used in conjunction with the circular Gabor filter equation (6-5) to generate a spatial pyramid of 10 scales. Note that the point of equation (6-5) is that the angle of the Gabor filter orientation is not necessary since edge detection occurs over all orientations. As in previous work [189] the free parameter σ is set at 4.5 and the template size of G is 11x11 pixels.

$$I = \frac{(r + g + b)}{3} \tag{6-4}$$

$$G(x, y) = \exp\left(-\frac{X^2 + Y^2}{2\sigma^2}\right)\cos\left(2\pi\sqrt{X^2 + Y^2}\right) \tag{6-5}$$

Subsequently, equations (6-6) and (6-7) below are executed in order to form the orientation conspicuity maps. Symbol Θ refers to across scale differences, symbol $\oplus$ to the across scale addition and *N* stands for normalisation. The *c* (centre) scales 1-3 are subtracted from scales 4-10 in *s* (surround) and this leads to a total of 7 feature maps for the three centre scales which are in turn (equation (6-7)) normalised and summed.

$$O(c, s) = O(c)\Theta O(s) \tag{6-6}$$

$$\overline{O} = \sum N\left(\bigoplus_{1}^{7} N(O(c, s))\right) \tag{6-7}$$

At the same time, double-opponency colour channels are formed according to equations (6-8) (6-9) (6-10) and (6-11). The circular Gabor filter and the double-opponent channels are operations that have been proven to exist in cone cells in the retina of biological organisms [221], [222].

$$R = r - \frac{(g + b)}{2} \tag{6-8}$$

$$G = g - \frac{(r + b)}{2} \tag{6-9}$$

$$B = b - \frac{(r + g)}{2} \tag{6-10}$$

$$Y = r + g - 2(|r - g| + b) \tag{6-11}$$

The double opponent channels *R, G, B* and *Y* are, similarly to the Gabor conspicuity maps, processed using the same number for centre and surround scales. The across scale differences of these channels, equations (6-12) and (6-13) then lead to 7 double-opponent colour conspicuity maps which are in turn normalised and summed, as shown in equations (6-14) and (6-15). Note that the *O, RG* and *BY* maps are treated exclusively in contrast to previous work [156], [163], [189], [197], as this particular summation of features into one universal saliency map has neither been proven to exist from physiological studies nor shown any significant computational advantage other than visual

representation. Furthermore, the summation of features raises questions about the priority or the internal relationships between different features which have not been addressed.

$$RG(c,s) = \left(R(c) - G(c) \Theta G(s) - R(s)\right) \quad (6\text{-}12)$$

$$BY(c,s) = \left(B(c) - Y(c)\right) \Theta \left(B(s) - Y(s)\right) \quad (6\text{-}13)$$

$$\overline{C}_{RG} = \sum N\left(\bigoplus_{1}^{7} N(RG(c,s))\right) \quad (6\text{-}14)$$

$$\overline{C}_{BY} = \sum N\left(\bigoplus_{1}^{7} N(BY(c,s))\right) \quad (6\text{-}15)$$

After normalisation and across scale addition, lateral inhibition is applied to promote the contrast between low and high values in the map, following physiological studies which suggest the presence of such mechanism [57]. Additionally, a slight blur using a Gaussian (5x5, σ = 4) is used in order to broaden the saliency areas. The final saliency maps highlight the most prominent ROI in the image and are used as topographic maps of salient feature information that is later used for recognition. A lateral mechanism is also applied at this stage to suppress background “noise” and promote salient regions. This mechanism follows equation (4-44) and instead of setting the values below R to zero, they are reduced by 50%.

The processes of converting the image to intensity, Gabor spatial pyramids, colour-opponent channels and colour pyramids are shared with the recognition part of the algorithm as can be seen in the following section. Unlike past studies in which salience would feed recognition sequentially with no common processes and interaction, here, salience and recognition share common processes up to the formation of Gabor pyramids. This follows from knowledge of the biological process, since the ventral and dorsal visual pathways split after the V1 (i.e. where edge detection occurs).

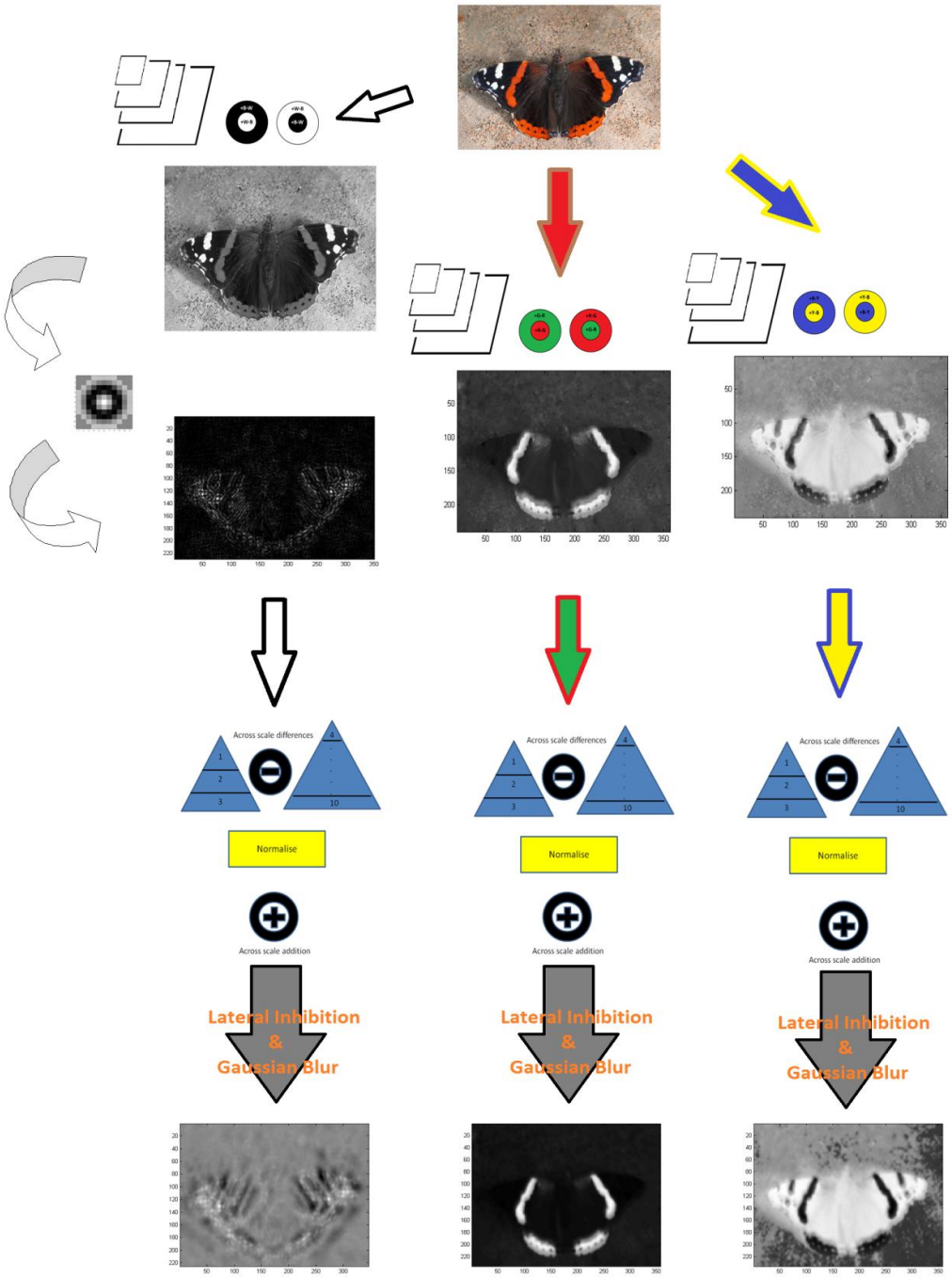


Figure 6-11: The illustrated extraction method for saliency maps. A pyramid for edge detection from intensity, a Red-Green pyramid and a Blue-Yellow are across-scale subtracted and after normalisation across-scale added. The final saliency maps are produced after a lateral inhibition mechanism and Gaussian blur. The steps before the across-scale differences stage, is shared with the recognition part of the algorithm (ventral stream).

6.2.2 Colour and shape in biologically-inspired object recognition

Colour feature extraction is achieved from the centre-surround opponent colour difference distribution, progressively max-pooled with image pyramids. More specifically, the process between equations (6-8) to (6-13) is identical and shared with CUVS. This process as illustrated in Figure 6-11 is identical until the normalisation step. To preserve the non-linear distribution of differences between the opponent channels, and because there is no intention of summing dissimilar features into one map as in saliency, normalisation is not employed for the following equations (6-16) and (6-17) as shown below:

$$RG_{tot} = \sum \left(\bigoplus_{1}^{7} (RG(c,s)) \right) \tag{6-16}$$

$$BY_{tot} = \sum \left(\bigoplus_{1}^{7} (BY(c,s)) \right) \tag{6-17}$$

After obtaining representations for RG_{tot} and BY_{tot}, two pyramids are respectively created each having 10 scales and treated similarly to the Gabor pyramid in previous sections. As before the max-filter pyramid of two scales, moves with a subsampling factor of 2 and in steps of 5 pixels along both directions. In this way, the maximum values of the colour distribution in the two opponent pyramids provide a kind of colour-spatial signature of the object in the scene. Executing the recognition procedure similarly to section 6.1.2, the original opponent channels now condense into S2 maps from which feature templates can be extracted.

At this stage of the research it is difficult to hypothesize the percentage of features required out of the three different pyramids *C1, RG* and *BY*. There is likely a task and/or dataset dependent equation between them which would more accurately capture their exact relationship. In this work, the choice between features i.e. *C1* or *RG* or *BY,* has been set to random and therefore varies between experiments. After a particular number of features for the 3 pyramids has been extracted, all features are then stored in the same shared featurebook. To address the different range of values in vectors from *C1, RG, BY*, the values for σ in equation (6-5) has been adjusted to 4.5.

The algorithm is first tested with the baseline FHLib-like parameterisation (MFHLib) which requires 140 pixels for an image's short edge, 4 different size patches (4x4, 8x8, 12x12, 16x16) and 11x11 circular Gabor filter banks. In other words, the setup is identical to [189]. Subsequently, the algorithm is enhanced by introducing a higher resolution for each image (240 pixels, short edge), adding four more patch sizes 20x20, 24x24, 28x28, 30x30 and retaining the one

circular Gabor filter for edge detection. With these added enhancements the feature extraction method is then attached using saliency. Finally, the *S3* and *C3* layers are in turn embedded with a feature library of 100% (the amount of features retained after cleaning the library from repeated features), this is the algorithm SFHLib from section 6.1.2 which operates only on the morphological features of the objects in images.

The second step in the experiments is to introduce the colour features. The Cranfield/Colour/Cortex-like Object Recognition (COR) refers to SFHLib with the addition of colour recognition. COR is only executed with a feature library percentage of 100% in this work (COR100) since feature reduction is beyond the scope of this experimental analysis and has been examined in section 6.1.1.3. There are two versions of COR100 being considered, COR100 in which the salient features are extracted from morphological salience maps only and COR100 (m+s) in which salient feature extraction relies on both morphological and spectral salience maps. Each saliency map matches the size of the original image being used in object recognition exactly i.e. 240 pixels for the shortest edge. For all experiments during training, an abundant number of features were chosen. Specifically, for all datasets, 30 images for each class were chosen for training and a different set of 30 images per class for testing. This means that although in each dataset the total number of features in the featurebook is varied i.e. 6000 features for the 4 class datasets, 9000 for six class datasets, 15000 for the 10 class and 37500 for the 25 class, the number of features per image is always 50 (this particular criterion is identical to chapter 6).

For the initial set of experiments the classifier is treated as constant. The parameters of the linear SVM being used have been found using the cross-validation technique. In order to further testify improvement in performance, the methodologies need to be compared against different classification schemes. In particular, two additional methods are considered here, an SVM with a RBF kernel and a “Gentle” Adaboost classifier with 100 weaklearners (perceptrons at 1000 iterations with $\lambda=0.001$ and $\varepsilon = 1$). RBF kernel parameter values *C* and *gamma* have been found similarly to the linear classifier, using cross-validation. An analytical table of their values is provided in Table 6-10. The algorithm layout is depicted in Figure 6-12 and its steps explained in more detail below:

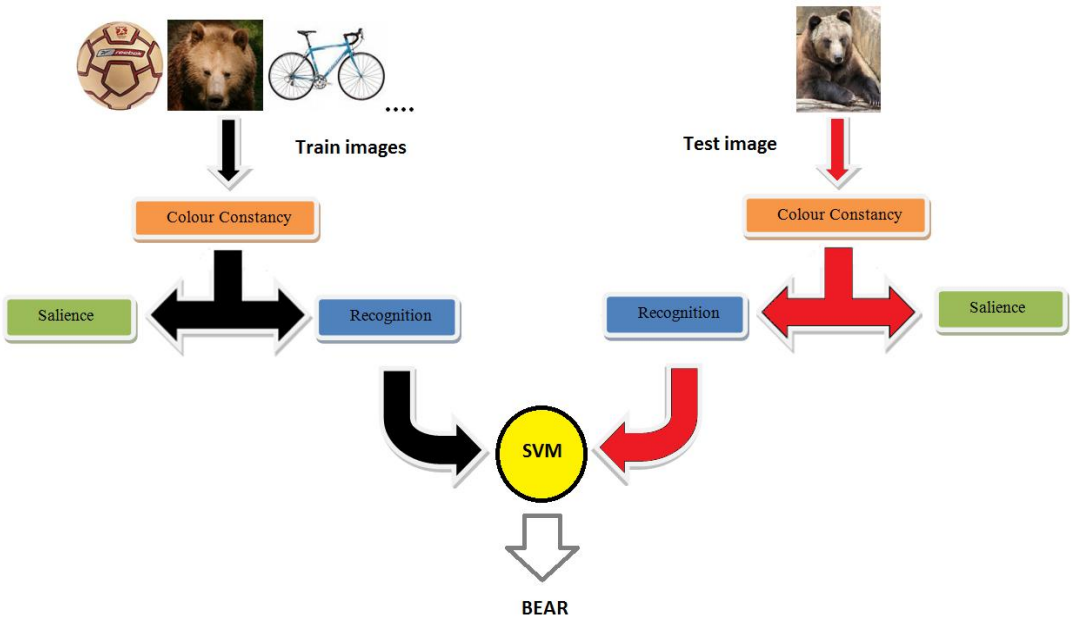


Figure 6-12: The layout of the algorithm.

1. Train path (black route **Figure 6-12**) receives the input train images.
2. Use colour constancy to obtain the spectral information of objects.
3. Apply edge detection with the circular Gabor filter.
4. Perform morphological only salience (SFHLib, COR100) or morphological and spectral salience (COR100 m+s) according to section 3.1.
5. Isolate salient ROI within the object then algorithm extracts S2 templates from these areas following the procedure explained in section 6.1.1.
6. Max responses of all S2 templates produce C2 vectors which in turn train the classifier.
7. In the testing stage (red route in **Figure 6-12**) steps 2 and 3 are identical.
8. Salient ROI coordinates within the object are retained. Template matching is performed over known salient areas only. Naturally, in COR100 (m+s), colour salient areas are also included.
9. The C2 testing responses are fed into the same classifier to produce the recognition results.

As before, MFHLib is the building block for all improvements of this section. SFHLib results are provided in Table 6-8 to illustrate the performance difference between the early morphological-only recognition approach and the versions of the algorithm developed for this study.

From Table 6-8, it can be seen that the COR100 algorithm with morphological + spectral salience performs marginally better, followed by COR100 with morphological salience only. The highest improvement by almost 32% is observed against the MFHLib 10 class dataset result. Importantly, the three datasets Butterflies, Birds, and Cats, exhibit a consistently improved performance over the original model while having similar classification accuracy scores. The trend of decreased performance in the multiclass datasets (CUUD, CUCD, 10 class and 25 class) as the number of classes increases is observed in Table 6-8. The worst results overall have been produced by MFHLib indicating its weaknesses as a model to cope with more complex classification tasks. An important conclusion from Table 6-8 is that even though the multiclass datasets do not share a particular colour pattern between their classes i.e. cars have many different colours, colour recognition enhances performance.

Method / **Dataset**	MFHLib	SFHLib 100%	COR100 morphological salience	COR100 with morphological + spectral salience
Butterflies	47.5	45.1	48.4	52.3
Birds	49.8	46.6	52.7	52.8
Cats	47.5	47.3	50.1	51.9
CUUD	72	82.2	85.5	86.3
CUCD	64	66	74.2	73.9
10 class	28	53	58.5	59.8
25 class	23	36	41.5	41.3

Table 6-8: Average percentage classification accuracies over 3 independent runs for the seven datasets. All results typically vary at ±1.5%).

Method / **Dataset**	MFHLib	SFHLib 100%	COR100 morphological salience	COR100 with morphological + spectral salience
Butterflies	13.2 (σ) 8.6 (MAD)	13.1(σ) 5.9 (MAD)	12.3 (σ) 2.7 (MAD)	7.3 (σ) 3.8 (MAD)
Birds	10.8 (σ) 7 (MAD)	8 (σ) 7 (MAD)	8.2 (σ) 4.3 (MAD)	8.6 (σ) 5.4 (MAD)
Cats	17.6 (σ) 12.4 (MAD)	20.2 (σ) 12.4 (MAD)	11.4 (σ) 2.1 (MAD)	10.3 (σ) 8.6 (MAD)

Table 6-9: Standard deviation (σ) and Median Absolute Deviation (MAD) results for the three colour and morphology based datasets.

By examining the standard deviation (σ) and median absolute deviation (*MAD*) for the “Butterflies” and “Cats” datasets in Table 6-9, it is further noticeable that the COR100 algorithms substantially improve the variability of the results compared to the previous versions MFHLib and SFHLib. This tendency is also

apparent with the “Birds” dataset for the MAD values and to a smaller extent for the standard deviation values.

Another key aspect of the new algorithm’s behaviour is the feature reduction process. In the left bar plots of Figure 6-13 and Figure 6-14, the random choice of features between the three pyramids yields approximately equal percentages over the large pools of features of 9000 for Figure 6-13 and 15000 for Figure 6-14. So for example, the left *C1* pyramid in Figure 6-13 with a percentage of 33% is 2970 features large. After inserting layers *S3/C3*, the removal of duplicate features changes the total numbers to 8430 and 13360 features. Furthermore, the percentages change slightly with respect to their original feature percentages. There is no pattern between the feature reduction percentages since they are controlled from the salience process and in essence the dataset on which this process is applied.

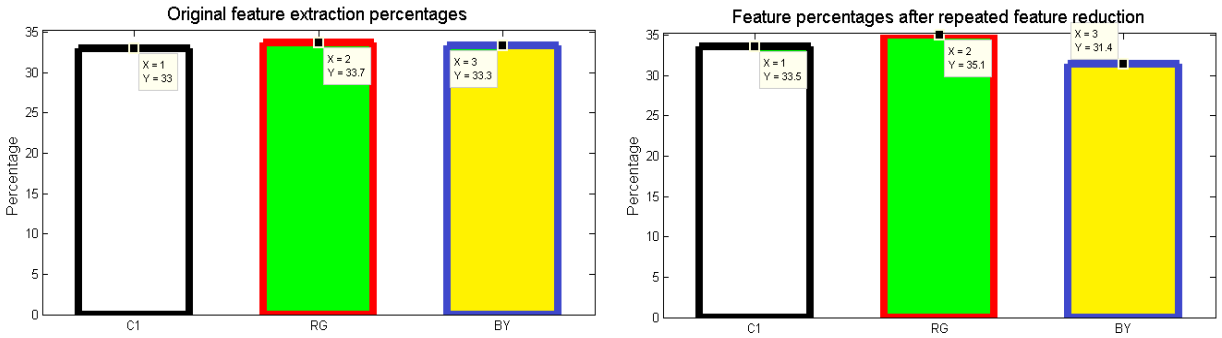


Figure 6-13: The first run of the COR100 (morphology + spectral salience) algorithm on the “butterflies” dataset. The bar plot on the left shows the percentages over the total number of 9000 features shared in each pyramid i.e. C1 for morphology, RG for red-green and BY for blue-yellow. The bar plot on the right shows the percentage after repeated feature reduction.

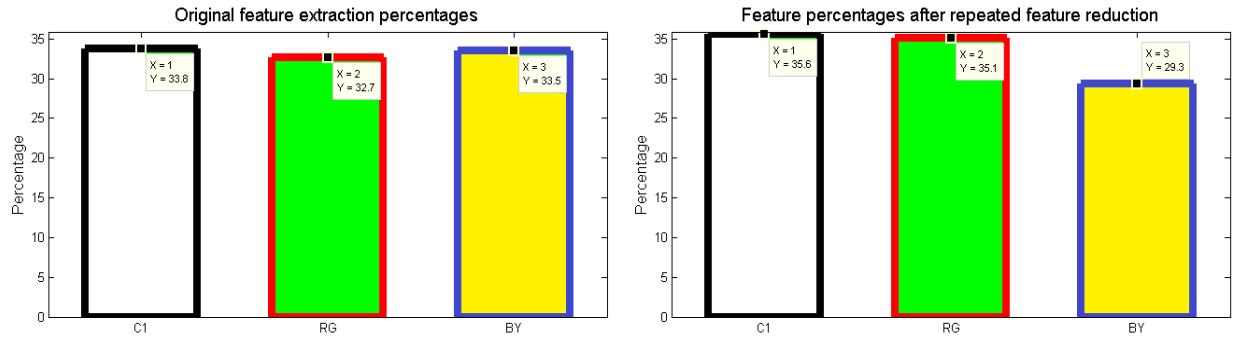


Figure 6-14: The first run of the COR100 (morphology + spectral salience) algorithm on the “10class” dataset. The bar plot on the left shows the percentages over the total number of 15000 features shared in each pyramid i.e. C1 for morphology, RG for red-green and BY for blue-yellow. The bar plots on the right shows the percentage after repeated feature reduction.

In addition to validating the newly developed COR100 and COR100 (m+s) algorithms with different datasets, it is imperative to assess their behaviour against different classifiers as in Table 6-11. Table 6-10, shows the best classification values found for parameters *C* and *gamma*, through the cross-validation technique for three SVM kernels.

Dataset / Kernel	Butterflies	Birds	Cats
Linear			
C	∞	∞	∞
gamma	1	1	1
Sigmoid			
C	100	100	10
gamma	Def	Def	Def
RBF			
C	100	100	1000
gamma	Def	Def	Def

Table 6-10: The best values of C and gamma for each kernel as found via cross-validation for each dataset separately. (Default - Def. = 1/number of features)

In Table 6-11 below, the superior performance of the COR algorithms with respect to SFHLib is generally exhibited across all classification methods. As an exception, under the RBF kernel for the “birds” dataset, SFHLib shows better results but this result is not reproduced from the other 3 classifiers indicating an anomaly introduced from the specific classifier.

In this table the Gentle Adaptive Boosting (Adaboost) classifier is introduced. In general, Adaboost is a learning algorithm which linearly combines “weak” classifiers, e.g. simple perceptrons. The final decision is given by [223]:

$$H(x) = sign\left(\sum_{t=1}^{T} a_t h_t(x)\right) \qquad (6\text{-}18)$$

In equation (6-18), *T* is the series of rounds, *α* is the weight of each classifier and *h* is the weak hypothesis. The weight α is defined as [223]:

$$a_t = 0.5\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \qquad (6\text{-}19)$$

The weak classifier attempts to find a hypothesis *h*, whose performance is measured by an error (ε_t) with respect to the distribution of training vectors. Additionally, the error is given by [223]:

$$\varepsilon = \frac{\sum_{t=1}^{N} w_i I(y_i \neq h_t(x_i))}{\sum_{i=1}^{N} w_i} \quad (6\text{-}20)$$

N is the number of different combinations. The weak classifiers are trained in series and the weight of classified combinations with high error is increased while for more accurately classified combinations the weight is decreased. The weights are initialised at [223]:

$$w_i = \frac{1}{N}, i \in \{1,...,N\} \quad (6\text{-}21)$$

The Gentle Adaboost variant in particular, optimises the cost function below, by using weighted least-squares regression:

$$E[\exp(-yH(x))] \quad (6\text{-}22)$$

With the "butterflies" and "birds" dataset the COR100 (m+s) method is marginally worse in contrast to Table 6-8, a behaviour which is slightly enhanced with the Adaboost classifier. Results indicate that better performance between the two COR100 methods is dataset-dependent. It is further evident from Table 6-11 that the "Gentle" Adaboost classifier captured the COR algorithms better than the other classification methods and given its simplicity in parameterisation, it is therefore used in the next sections.

Dataset / Method	Butterflies (%)	Birds (%)	Cats (%)
SFHLib 100% Sigmoid	44.3	44.6	37
COR100 - SVM Sigmoid	50.2	51.2	46.7
COR100 m+s - SVM Sigmoid	49.1	50.5	48.6
SFHLib 100% RBF	42.8	51.2	49.9
COR100 - SVM RBF	49.8	48.9	48.7
COR100 m+s - SVM RBF	50.1	46.2	50.9
SFHLib 100%Adaboost	47.8	44	40.5
COR100 - Adaboost	58.4	57.9	50.5
COR100 m+s - Adaboost	56.3	56.3	52.5

Table 6-11: Classification accuracies under different classification techniques as an average over 3 independent runs on the Butterflies, Birds and Cats datasets. All results typically vary at ±1.5%.

6.2.3 Section Conclusions

The SFHLib algorithm was enriched with a proposed double-opponency mechanism that learns colour features alongside the standard morphological features. With a model that investigates both morphology and colour of objects in a biologically-inspired manner, two versions of salient feature extraction were examined in order to determine their relationship against classification accuracy. In particular, COR100 is the termed version of the algorithm which focuses on morphological-only salient features while extracting both morphological and colour features for recognition. The COR100 (m+s) version examines both morphological and colour salient features while similarly to COR100 uses morphological and colour features for recognition. Both versions of the algorithm produced enhanced results with respect to previous MFHLib and SFHLib models of morphological-only recognition and this performance has been consistent across all of the used datasets. With respect to MFHLib the use of colour for recognition has doubled the classification accuracy in some cases and has even been shown to enhance the performance for classes that do not demonstrate a particular colour pattern. The new models were also validated for improved classification accuracy under different classification schema. Overall, the results further indicate that choosing between the two versions of the algorithm depends on the object class and/or dataset but further experimentation with a greater variety of datasets and scenarios would be needed to fully validate this.

7 CONSTANCY AND TEXTURE

7.1.1 Colour Constancy application and comparison

In addition to the centre-surround and colour opponency processes examined in section 3.2, the human retina in conjunction with other parts of the visual system (lateral geniculate nucleus, V1) exhibits a phenomenon called colour constancy. Under colour constancy, detection of the colour appearance of an object is achieved despite illumination variations of the ambient light [64], [65]. Illumination variation can pose a serious problem to object recognition [224].

Coupling colour constancy with object recognition is not a new idea but introducing colour constancy to a biologically-inspired object recognition model, is a first here. Generally, merging colour constancy with recognition in the past provided contradicting results. Early results [225] showed enhanced recognition results with colour constancy. Other results [226], [227], failed to show whether colour constancy could improve recognition. Moreover, colour constancy was applied with conventional object recognition for outdoor scenes without explicitly showing any improvement [228]. Further research [229] illustrated that colour constancy should be applied with object recognition under certain conditions in which the morphological information is also considered. More recently [230], using a SIFT-based system and a pixel-based approach showed that classification accuracies can be improved with the use of colour constancy.

In the primary visual cortex, chromatic adaptation is accomplished with the cone cells transmitting the amount of black or white dilutions in colours and rods the intensity of ambient light. This human vision trait portrays the remarkable ability to disassociate intensity and/or dilutions from colour effortlessly, while preserving much of the actual spectral information of an object. This illumination-invariant characteristic is crucial because it makes biological spectral vision dynamic, i.e. performing consistently under a wide range of light changes.

It is known from section 3.3 that spectral perception for a given surface, depends on the illumination changes of the source as well as the reflectance characteristics of the surface itself. Therefore, colour representation cannot be therefore accurate without some form of colour constancy perception present in the light sensor. Such properties have been proven to exist in the human visual system and accurate artificial spectral representation of surfaces and objects for improved detection, segmentation and recognition algorithms is imperative.

To incorporate illumination invariance in the current biologically-inspired model, the images at the training and testing stages of the algorithm are simply

transformed using colour constancy prior to visual attention. The colour constancy models examined here are the popular SSR, MSR, Grey edge [231], Grey edge 2nd order [231], and maxRGB [68]. This process is illustrated in Figure 7-1.

Figure 7-1. Colour constancy example. Before the input image is used it gets processed from a colour constancy algorithm, in this example max-RGB. Note that the procedure here is outlined for illustrative purposes and colour differences between images may be visually difficult to distinguish.

Table 7-1, summarises the results obtained after 3 independent runs for each of the five colour constancy methods, under the two different algorithms and for four datasets i.e. the three colour and morphology datasets plus a morphology-based dataset with a moderate number of classes. With respect to the baseline results of the COR100 algorithms (first two rows of the table) without colour constancy, the reduced performance for the Retinex algorithms is obvious across all datasets. Marginally better average classification accuracies are seen for the Grey edge and maxRGB methods. The values highlighted in red are the highest values received for each dataset and it is noticeable that for each dataset there is a different combination of the algorithm maximally responding. Furthermore, it can be seen together from Table 6-11 and Table 7-1 that the choice between the morphological only and morphological with colour salience for the feature extraction process again depends on the dataset. Even so, only minor fluctuations in performance can be seen and in many instances in Table 7-1 within the variance percentage of 1.5%.

It is perhaps not obvious that employing colour constancy would improve performance from Table 7-1 as opposed to introducing colour within the recognition process as in Table 6-8. One reason could be that all image datasets are mostly in natural outdoor scenes and have relatively small fluctuations in illumination. Nevertheless, promising results here encourage future research that incorporates a greater number of datasets, colour

constancy techniques and calibrated cameras. In the next section the employment of an adaptive method to estimate the best colour constancy method for each image individually [96] is presented rather than the one colour constancy method for all as presented in this section.

Dataset / Method	Butterflies	Birds	Cats	10 class
COR100	58.4	57.9	50.5	59.5
COR100 m+s	56.3	56.3	52.5	59.3
COR100 SSR	43.7	42.6	32.8	44.4
COR100 m+s - SSR	44	40.3	40	43.1
COR100 MSR	43.7	42.5	36.5	43.6
COR100 m+s - MSR	43.5	38.8	41.5	46.2
COR100 Grey Edge	58	56.8	50.5	58
COR100 m+s – Grey Edge	58.2	54.5	55.4	57.5
COR100 Grey Edge 2nd	59.3	54.3	52.9	57.3
COR100 m+s – Grey Edge 2nd	58.6	53.9	55.4	60
COR100 maxRGB	55.5	57.2	52	60.5
COR100 m+s - maxRGB	59.3	55.4	52.3	59.7

Table 7-1: Average percentage classification accuracies over 3 independent under five colour constancy techniques using the COR100 algorithm on the Butterflies, Birds and Cats datasets. All results typically vary at ±1.5%.

7.1.2 Synthesising Colour Constancy methods

Following on from the results of experiments shown in Table 7-1, in this section the fusion of colour constancy algorithms is applied using the theory and methodology from section 3.3.2. In addition to the original work in [96] where the angular error is used as the performance measure, the same methodology is evaluated as a whole using the classification accuracies of the object recognition model. Specifically, using a pre-trained SVM classifier with the datasets from 5.2.3, at the colour constancy step and for each image an adaptive decision is introduced before transforming the image (Figure 7-2).

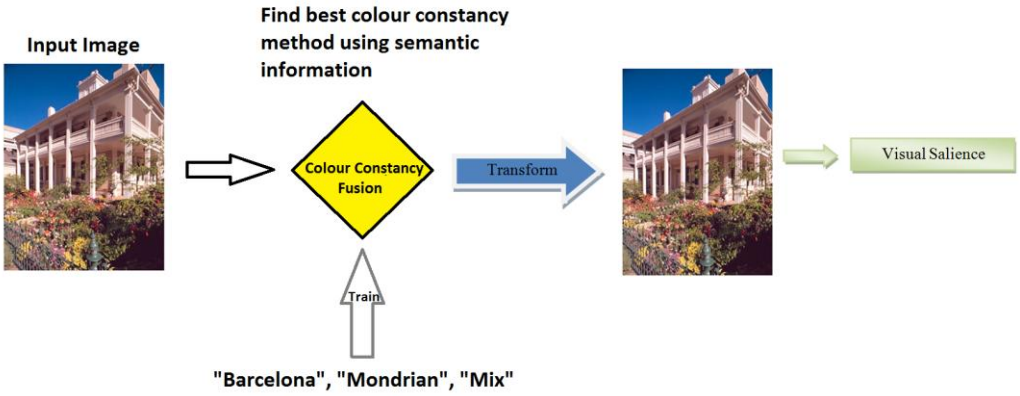


Figure 7-2: The simplified procedure of experiments in this section

Ideally, the SVM classifier in colour constancy fusion should be trained with the images of the evaluation datasets. However, ground truth illumination data for the datasets in section 5.2.2 is not available (in general, datasets with illumination ground truth data are scarce). Instead, Figure 7-2 illustrates how the datasets from section 5.2.3 were inserted into the current methodology. The list of colour constancy algorithms combined here is different from section 7.1.1, in that SSR and MSR, given their poor performance (Table 7-1) are replaced with the Shades of Grey [232] and Grey world [77] algorithms. So the methods examined in this section are:

- Shades of Grey
- Grey Edge
- Grey Edge 2nd order
- Grey world
- maxRGB

Table 7-2, Table 7-3 and Table 7-4 below were obtained with a similar setup as in section 6.2. A number of 9000 features was chosen for the datasets “Butterflies”, “Birds” and “Cats” and 15000 features for the 10 class dataset. Moreover, 30 images per class were chosen for the training step and 30 images per class for testing. The superiority of the COR100 algorithms was shown in the section 7.1.1 and given the same setup here, the tables can be directly compared with the previous section.

Dataset / Method	Butterflies	Birds	Cats	10 class	Mean over datasets
COR100 – Adaboost	58.1	54.3	50.9	57.8	55.3
COR100 m+s - Adaboost	57.2	55.5	52.1	57.3	55.5

Table 7-2: Average percentage classification accuracies over 3 independent runs for Colour Constancy fusion in the COR100 algorithms as obtained with the "Mondrian" dataset.

Dataset / Method	Butterflies	Birds	Cats	10 class	Mean over datasets
COR100 – Adaboost	57.2	55.9	49.5	57.3	
COR100 m+s - Adaboost	57.7	52.5	50	54	53.5

Table 7-3: Average percentage classification accuracies over 3 independent runs for Colour Constancy fusion in the COR100 algorithms as obtained with the "Barcelona" dataset.

Dataset / Method	Butterflies	Birds	Cats	10 class	Mean over datasets
COR100 – Adaboost	53.2	49.1	50.1	56.8	52.3
COR100 m+s - Adaboost	57.2	50.3	50.7	56.4	53.6

Table 7-4: Average percentage classification accuracies over 3 independent runs for Colour Constancy fusion in the COR100 algorithms as obtained with the "Mix" dataset.

By directly comparing the results of all datasets in Table 7-2, Table 7-3 and Table 7-4 with Table 7-1, no noticeable improvement in classification accuracy can be observed. Overall, there is a marginal decrease in performance, more evident in Table 7-3 and Table 7-4. Between the tables of this section, the highest classification accuracies were obtained from the "Mondrian" dataset. Importantly, the difference of classification accuracies between the tables signifies the impact the training colour constancy datasets have over the process. Another important observation is that the two COR100 versions of the algorithm again show as in the previous section, a dataset dependent behaviour.

Tables in this section focus on finding an increased and/or consistent object recognition performance. However, in the future an extensive comparison between the algorithms requires an additional photometric analysis, similar to [233], for all datasets individually.

7.1.3 Section Conclusions

In this preliminary study the main contribution has been the integration of colour constancy with bottom-up biologically-inspired object recognition for the purposes of illumination invariance for the first time. Before reaching this stage, the existing SFHLib algorithm was enriched with a double-opponency mechanism (section 6.2) that learns colour features alongside the standard morphological features. With a model that investigates both morphology and colour of objects in a biologically-inspired manner, two versions of salient feature extraction were examined in order to determine their relationship against classification accuracy. Colour constancy was introduced via five popular methods. Generally, the Grey-Edge and maxRGB methods showed better results over the Retinex models and in most cases marginally outperformed the baseline COR100 models. Results on illumination invariance have shown that it is difficult based on the five colour constancy methods and the four datasets examined, to estimate the optimum adaptive mechanism for varying illumination conditions. It was imperative therefore to investigate adaptive methods that estimate the best colour constancy method for each image individually rather than one colour constancy method for all as in section 7.1.1.

In the following section 7.1.2, the colour constancy fusion tables have not shown any improvement over the method presented in section 7.1.1 but have illustrated the need for an improved adaptive illumination mechanism, an additional photometric analysis and the creation of an object recognition dataset with ground truth data. The adaptive mechanism examines each scene independently and using semantic scene information from a pre-trained classifier, it can deduce which colour constancy method best describes it. For the first time such an illumination invariant approach was investigated for biologically-inspired vision. Further experimentation should also expand on an extensive colour constancy dataset that encompasses a greater variety of scenes (i.e. city environments, indoor images etc), and preferably one that is also used by the object recognition mechanism. In addition, a greater number of colour constancy methods should be examined while an alternative biologically-inspired illumination invariant mechanism should also be considered.

7.2 Texture in cortex-like object recognition

Gabor feature parameters have so far been treated in this thesis as constant and even though their parameterisation as such is suboptimal, they can still produce reliable results. Gabor features in HMAX (section 4.2.3) cover a wide variety of tuning combinations to express the morphological and textural information of an image. According to the creators, the fixed sets of Gabor parameters in HMAX have been extracted from physiological experiments and remain fixed regardless of the scene's content. However, equation (4-34) is obtained from unspecified empirical data, making Gabor parameterisation in HMAX ambiguous. In FHLib (also in IKN and GBVS) as it has been seen in sections of this thesis so far, the set of parameters σ, λ and γ is constant. While this combination of parameters has been shown to describe morphological information well, it is far from practical and ideal, especially for fine texture or multiple texture representation. It is known from [109], [116] that simple cell selectivity varies considerably and adapts according to the task, rendering HMAX, FHLib, IKN and GBVS as oversimplified approaches.

Before HMAX another multiple filter channel model was proposed in [234]. This multiple filter model used a bank of Gabor filters to process the original image and subsequently with an unsupervised clustering approach to perform segmentation. This model is computationally expensive as the necessary number of filters increases. Furthermore, it does not show any adaptability to an image's context and it is questionable whether a complete description can be achieved. A different method was examined and improved in [235], in which Gabor features are tuned according to an "information diagram" for each filter's space. With this approach a 2D array is formed according to the maximum values to a particular combination of spatial frequency versus orientation. The information diagram was introduced in [105], [236] but as the authors acknowledge, this concept has the serious drawback of being reliable only in uncluttered environments. A more adaptive approach with genetic algorithms is followed in [237], [238] for face recognition. Specifically, the genetic algorithm finds the optimum face areas for recognition and then spatial frequencies and orientations are selected by a feature selection algorithm (Sequential Floating Forward Search - SFFS). This method produced higher than 80% classification accuracies but was only tested against a single dataset for faces. Another drawback is its complexity and thus time-consumption.

It is evident from past work that adaptive Gabor parameterisation still remains an active research area. Examining this particular topic more closely is expected to improve both texture recognition tasks and overall classification since texture can be a significant feature for the distinction between certain

classes and subcategories of objects. In addition, as previously mentioned it is known (section 3.4 and [105], [107], [109], [116]) that mammalian vision covers a great number of different spatial frequencies and orientation sensitivities with the presence of millions of simple cells in V1. It has also been proven that the grating cell Gabor operators that have been used in this thesis, perform better than any other Gabor filter feature [239] for texture representation. It is the intent of this section therefore, to explore and propose a new and alternative way of optimising Gabor parameterisation using the filter layers of the present model.

For general recognition tasks it is unknown, however, the exact contribution that texture has on the overall process of a bottom-up or top-down process. For this reason, it is entirely treated in this section as a separate feature to colour and shape, i.e. the constant set of Gabor parameters from FHLib, which expresses broader morphological object information.

7.2.1 Hardcoded Method

The first method in this work on Gabor parameterisation and texture representation simply expands the current algorithm from that of section 6.2 with a separate operation stemming out of the existing morphology (Figure 7-3) but with a separate set of Gabor parameters. This new texture feature channel is otherwise identical therefore, with the shape channel but there are a couple of differences between them:

a) The set of parameters in equation (6-1) is *1* and *8* for σ and λ respectively.

b) The max pyramid's lowest scale is changed to *4x4*, with a 2 unit step.

This set of parameters above was chosen as the highest average response of all values in the *S1* layer of all images in the datasets (CUUD, CUCD). This is termed hereafter as a hardcoded method since these values were found manually and fixed throughout the experiments. The hardcoded method is intended to provide an indication of how the algorithm behaves with a separate simplistically tuned feature channel for texture and perhaps highlight the necessity for exploring other methodologies.

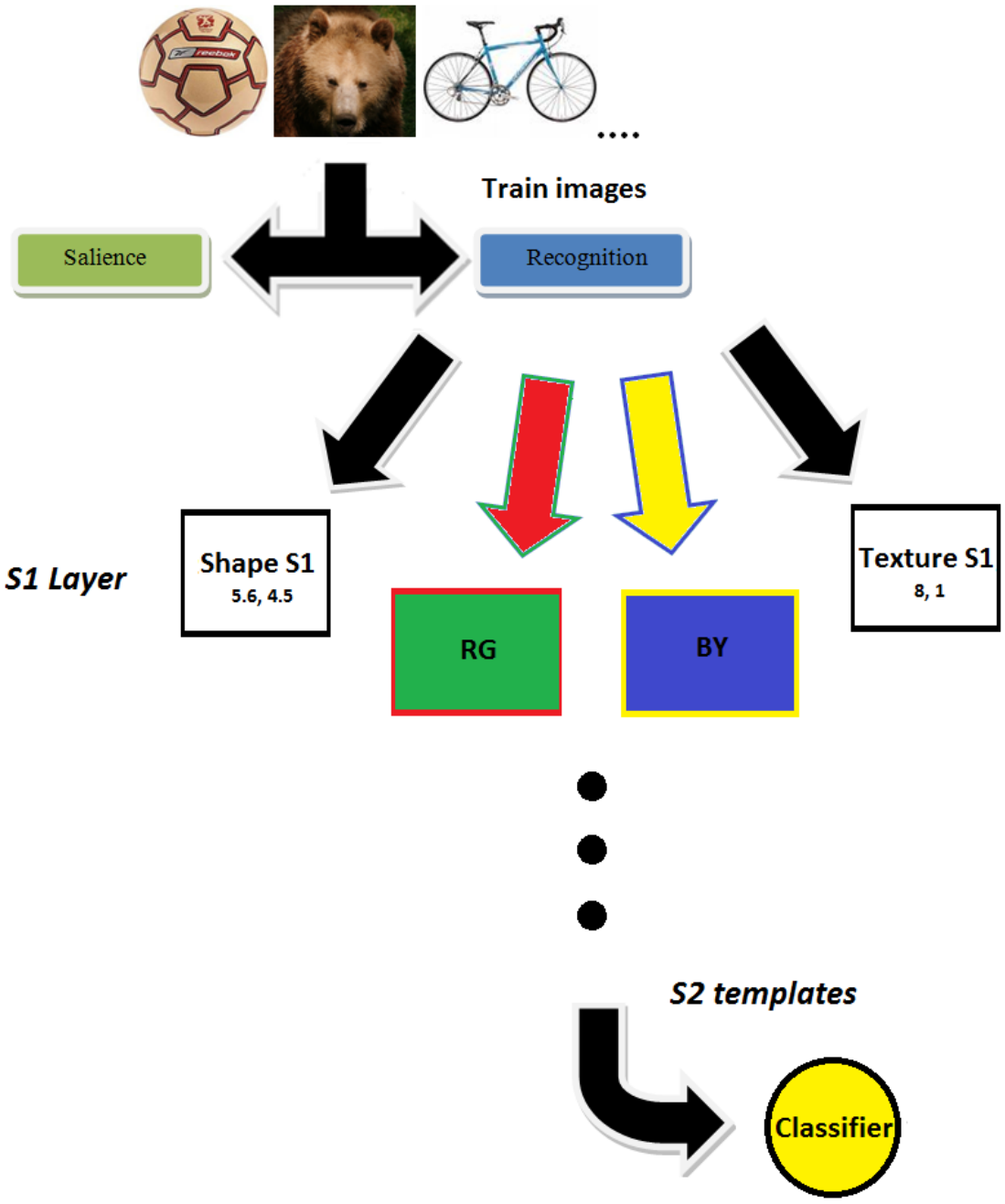


Figure 7-3: An illustrative overview of the major algorithmic steps under the hardcoded method during training. Texture (λ = 8, σ = 1) is treated as a separate feature to Shape (λ = 5.6, σ = 4.5)

7.2.2 Multiple Gabor Channel

The naive hardcoded method serves as comparison against more elaborate methods examined in this chapter, such as the multiple channel (or Gabor filter battery) approach inspired from [184], [234]. The main objective of this technique is to create a number of differently tuned texture channels with biological-like parameterisation, and combine them together as a texture

feature. In contrast to [184], biologically-inspired equations (3-14) and (3-15) are preferred over the unreferenced empirical (4-34) and (4-35). With respect to [234] there is a distinct intrinsic structural difference and the multichannel Gabor responses here are combined without any aid from statistical clustering algorithms.

It was mentioned in section 3.4.2, that according to physiological experiments the bandwidth of simple cells in V1 varies between the limits of 0.4 – 2.5 octaves [108] and the bulk of simple cells is between 1-1.8 with a rough average at 1.4. Therefore, the two equations (3-14) and (3-15) can be applied under this assumption. Another assumption made here is to set the standard deviation σ to 4.5, as in MFHLib. Ideally, this value is expressing effectively the size of the receptive field and should also be varied to accommodate the wide set of parameters present in V1. Nevertheless, there has to be a compromise for time-consumption here given the nature of the application in MATLAB, since each distinct parameter set adds extra computation penalties. Hence, only the spatial frequency via the wavelength λ is changed within the bandwidth range 1-1.8 (Table 7-5) with a 0.07 step in order to limit the result to 12 different combinations. This step again was chosen as a compromise similarly to the one made for σ but also because it produces noticeable enough variations in wavelength (Table 7-5).

Bandwidth (b) / Sigma	1	1.07	1.14	1.21	1.28	1.35	1.42	1.49	1.56	1.53	1.7	1.77
$\sigma = 4.5$	8.0	8.5	9.0	9.5	10.0	10.5	10.9	11.4	11.8	12.3	12.7	13.1

Table 7-5: The twelve wavelength λ values obtained by applying equations (3-14) (3-15) and used to set up the multiple circular Gabor filters.

The object recognition steps for the Multiple Channel approach are given below and illustrated in Figure 7-4:

1. During training calculate one shape (*S1*) and two colour (*RG, BY*) pyramids separately.
2. Calculate 12 texture pyramids by using multiple circular Gabor filters according to Table 7-5 and use the max pyramid with a *4x4* base, 2 unit step.
3. Find the maximum value for every unit in the *S1* layer of the 12 texture pyramids, in order to obtain one pyramid that contains the maximum response across all different texture values.
4. Use the texture pyramid as the fourth pyramid from which templates are extracted.

5. During testing calculate one shape, two colour and one texture pyramid as in steps 2-4. Use the stored templates from the featurebook on the respective pyramids to obtain *C2* vectors.

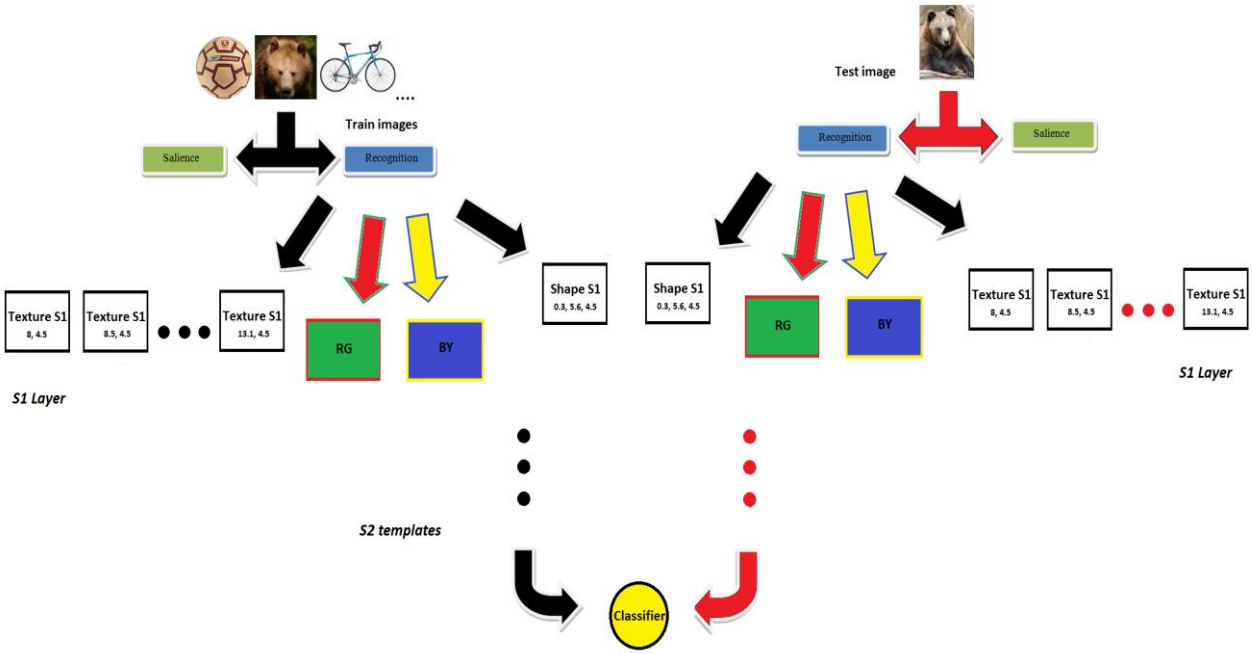


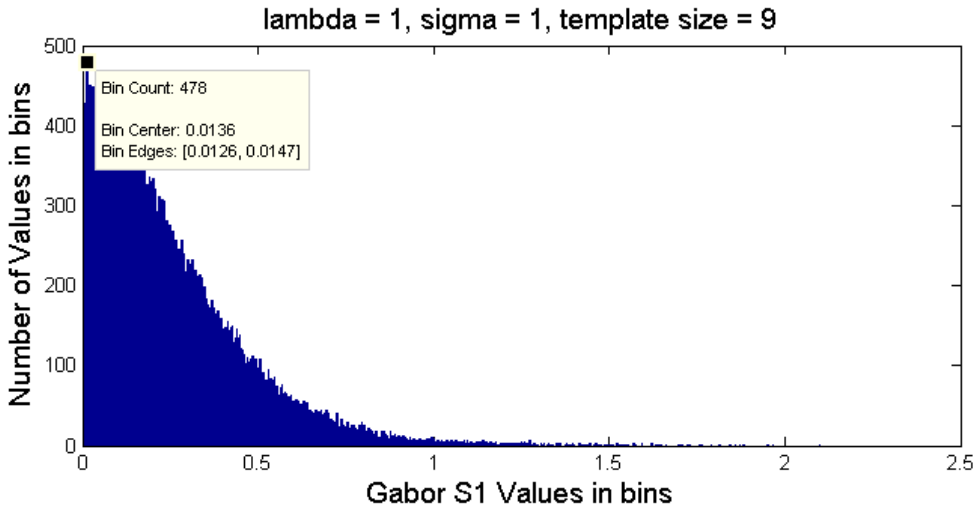
Figure 7-4: The layout of the algorithm when using the Multiple Gabor Channel method

7.2.3 Histogram Maximum Response

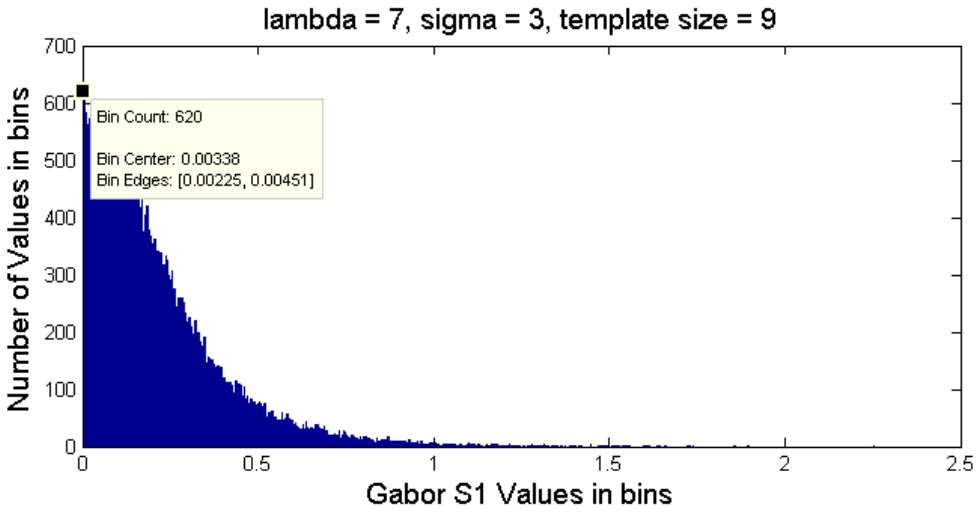
Hitherto, the Hardcoded and the Multiple Gabor channel methods address the problem in a similar manner, i.e. through Gabor processing. The Gabor processing techniques of the previous subsections may be faithful biological simulations but given the current computer resources, their serious limitations are parameter estimation and computation time.

The hypothesis here is that if values for σ, λ and filter size (N) are varied within a defined range, i.e. σ = {1, 3, 5, 7, 9}, λ = {1, 3, 5, 7, 9}, N = {9x9, 11x11, 13x13}, then the combination with the highest overall *S1* unit response, using a histogram, is the one picked out to express the texture feature pyramid (Figure 7-5).

(a)



(b)



(c)

Figure 7-5: (a) The input image, (b) the first histogram shown as an example (c) the "winning" histogram with the maximum peak response at 620 values. The values at the title in (c) are used to parameterise the circular Gabor filter.

A foreseeable drawback of this approach is that even though it applies to each image in the dataset individually, it takes background information into consideration indiscriminately or object information is expressed with just a single Gabor filter set of parameters. This technique is however, more adaptive and faster compared to Gabor processing methods. The method's steps are outlined below:

1. During training, calculate one shape (*S1*) and two colour (*RG, BY*) pyramids separately.
2. Establish the searchable range of Gabor parameter values, σ = {1, 3, 5, 7, 9}, λ = {1, 3, 5, 7, 9}, *N* = {9x9, 11x11, 13x13}.
3. Obtain Gabor responses only for the first scale of each image.
4. Generate histograms (1001 bins) of *S1* unit values for each combination of σ, λ, *N* and store the peak number of *S1* units from each histogram.
5. Use the maximum peak value of all histograms to form the texture feature pyramid using the respective set of parameters σ, λ and *N*.
6. Use the texture pyramid as the fourth pyramid from which templates are extracted.
7. During testing, calculate one shape, two colour and one texture pyramid as in steps 2-5. Use the *S2* stored templates from the featurebook on the respective pyramids to obtain *C2* vectors.

7.2.4 Gabor parameterisation using scene statistics

Inspired from the histogram maximum response technique for *S1* values and from [96], [99], [240], this subsection presents the implementation of a new method for Gabor parameterisation and thus texture representation. It is known [241] that Gabor texture features can be used efficiently for content–based and texture-based image retrieval tasks. The principle here is similar in that with the method described below, the texture "gist" of images as a whole, indicates which is the optimum set of Gabor parameters.

In particular, according to [99], the histogram of texture values in an image portrays a Weibull-like distribution and this concept was applied with Gaussian derivatives in [96]. Before the work from Geusebroek and Smeulders, a very similar approach was proposed using the Rician distribution for Gabor filter design in segmenting a two-texture image [240] The probability distribution function (PDF) of the two distributions is very similar. Another expression for the Weibull distribution can be retrieved by describing constant *C* in equation (3-3) in the following way for a variable *x*:

$$W(x) = \frac{1}{\beta}\left(\frac{x}{\beta}\right)^{\gamma-1} \exp\left(-\frac{1}{\gamma}\left|\frac{x}{\beta}\right|^{\gamma}\right) \qquad \textbf{(7-1)}$$

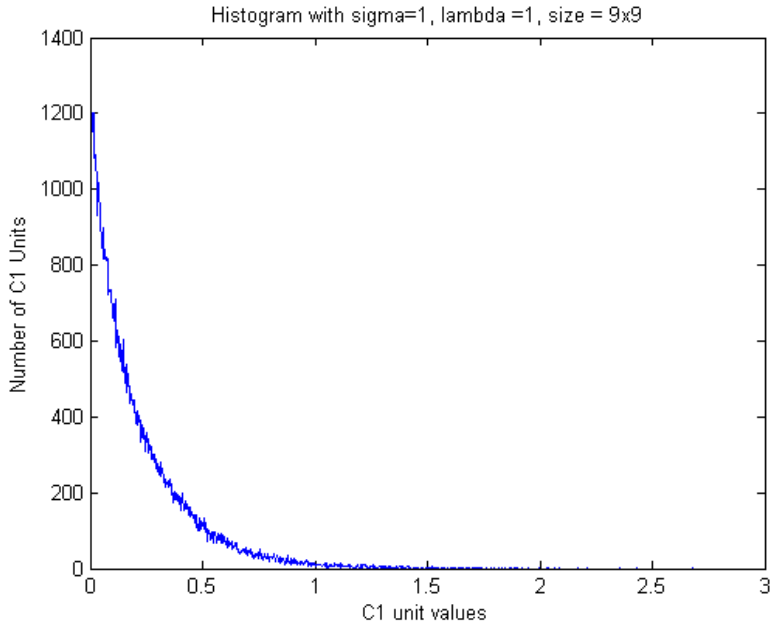
Equation (7-1) shows the exact relationship between parameters $\beta > 0$ (scale parameter) and $\gamma > 0$ (shape parameter), i.e. parameter β expresses the width and γ the amplitude of the distribution [96], for $\gamma = 2$ the Weibull distribution reduces to a Rayleigh distribution.

The Rician distribution on the other hand is given from the equation below [240]:

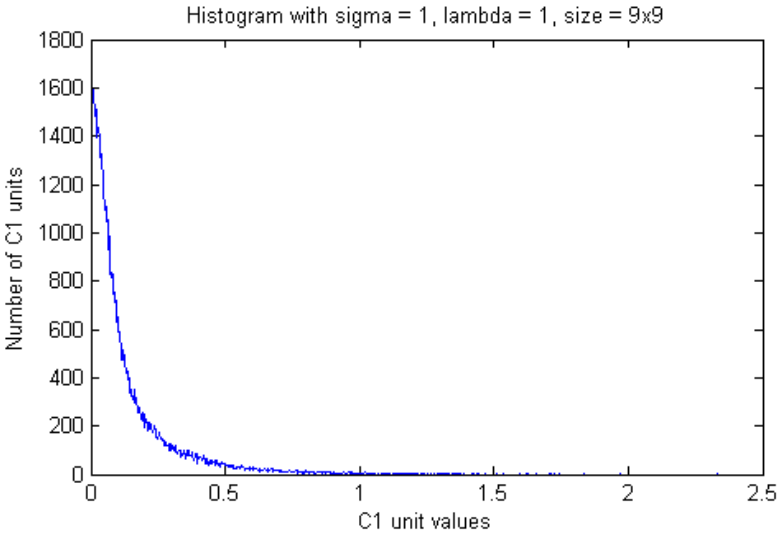
$$R(x) = \frac{x}{\sigma^2} \exp\left(\frac{-\left(x^2 + A^2\right)}{2\sigma^2}\right) \mathrm{I}o\left(\frac{xA}{\sigma^2}\right) \qquad \textbf{(7-2)}$$

In equation (7-2), $I_o(x)$ is the Bessel function of the first kind with zero order, A is the peak amplitude, σ the standard deviation and variable x is often referred as the magnitude or power. This distribution also reduces to a Rayleigh distribution when $A = 0$. The essential fitting difference between the Weibull and Rician distributions is the stretched tail seen in the Weibull especially as γ decreases, which matches the histogram of Gabor S1 values as seen in Figure 7-6 below:

(a) (b)



(c)



(d)

Figure 7-6: (a) and (b) are two example images of a ball and bear from the 10 class dataset, (c) and (d) are their respective S1 layer histogram plots with 1001 bins. The resemblance between them as well as the tail end of a Weibull distribution can modelled by values β and γ.

As in section 7.1.2, using the Maximum likelihood estimation with Newton's algorithm to integrate the image's histogram (30 iterations), yields the shape and scale parameters of the Weibull distribution. Each set of β and γ, found from varying σ = {1, 3, 5, 7, 9}, λ = {1, 3, 5, 7, 9}, template size = {9x9, 11x11, 13x13}, is stored for each image. After all images have been analysed then an SVM classifier is trained which is in turn stored and used during the testing stage to identify the best set of σ, λ and template size from the scale and shape of the histogram, following the test image's response to the same range of σ, λ and template size. The recognition algorithmic steps are outlined in more detail below and the general layout of the algorithm is depicted in Figure 7-7:

1. During training, calculate one shape (*S1*) and two colour (*RG, BY*) pyramids separately.
2. Establish the searchable range of Gabor parameter values, σ = {1, 3, 5, 7, 9}, λ = {1, 3, 5, 7, 9}, N = {9x9, 11x11, 13x13}.
3. Obtain Gabor S1 responses for the first scale only.
4. Generate histograms (1001 bins) of *S1* unit values for each combination of σ, λ and N.
5. Subsequently, match each histogram using Maximum Likelihood estimation (Newton's algorithm) with 30 iterations to a set of β and γ Weibull parameters. Every pair of β and γ, from each histogram is stored for every image in the same library.
6. Repeat for all images and store all values of β and γ to train an SVM classifier (RBF kernel, *gamma* = 1, C = 100, found from cross-validation)
7. During testing, calculate one shape, two colour and one texture pyramid as in steps 2-5.
8. Feed the testing β and γ pairs in the pre-trained SVM classifier and obtain best match of σ, λ and N. Then generate Gabor texture pyramid [242] and use the *S2* stored templates from the featurebook on the respective pyramids to obtain *C2* vectors.

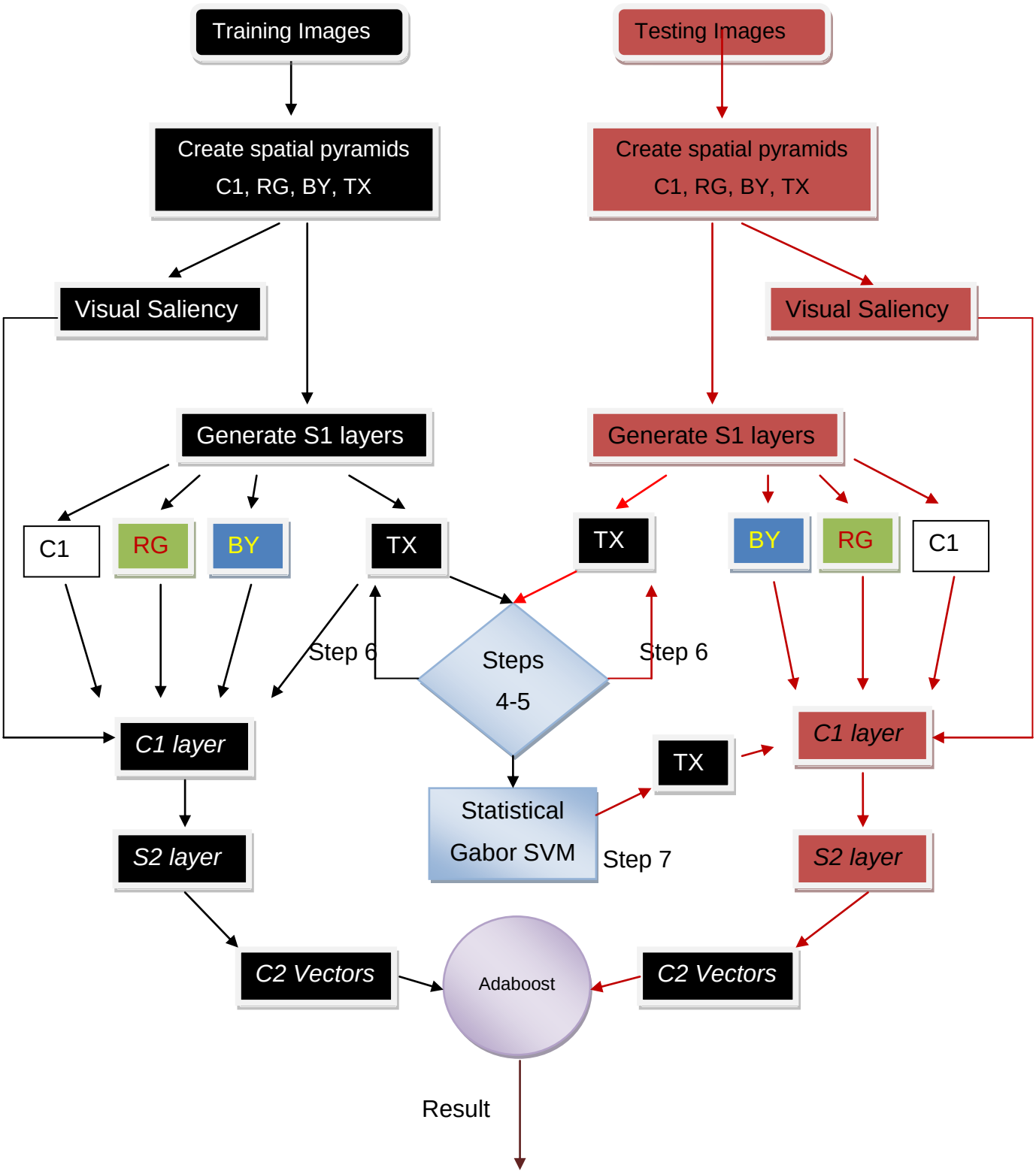


Figure 7-7: The general layout of the algorithm through statistical Gabor parameterisation.

7.2.5 Texture experiments

The methods described in subsections 7.2.1 to 7.2.4 have been developed with the complete absence of prior experimental evidence and comparison. For the first time here, four different Gabor parameterisation methods for texture recognition are evaluated together for the purposes of object recognition.

Six different datasets, “TX10”, “CUUD”, “Butterflies”, “Birds”, “Cats” and “10class” (section 5.2.2) were chosen. The choice of the datasets was significant, for the following reasons:

1. TX10 is a greyscale image texture database which has no distinct object boundaries.

2. CUUD is a naive small dataset without any background clutter information interfering with the recognition process.

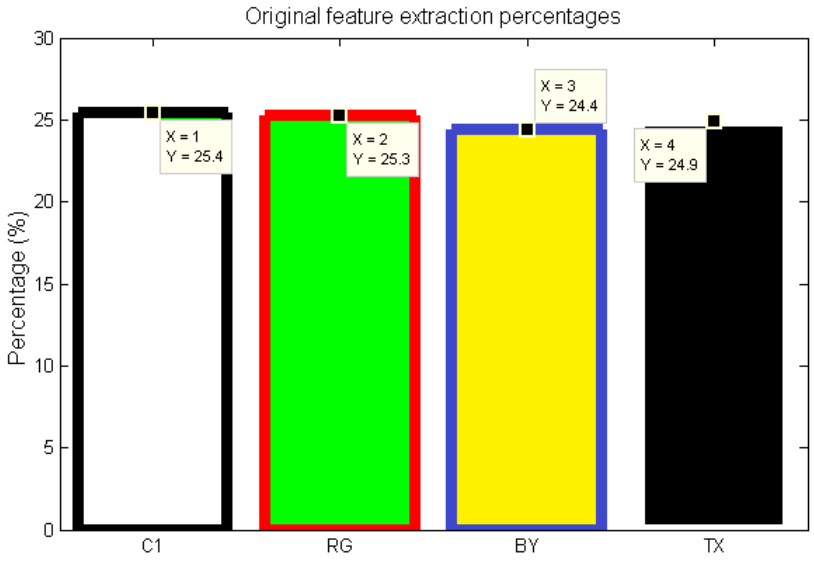
3. Butterflies, Birds and Cats portray animals of the same class category but of different subspecies that apart from colour are recognisable from texture.

4. The 10 class multiclass dataset does not show the same distinct patterns of colour and/or texture as the previous datasets but serves as a reference to a more general object recognition scenario.

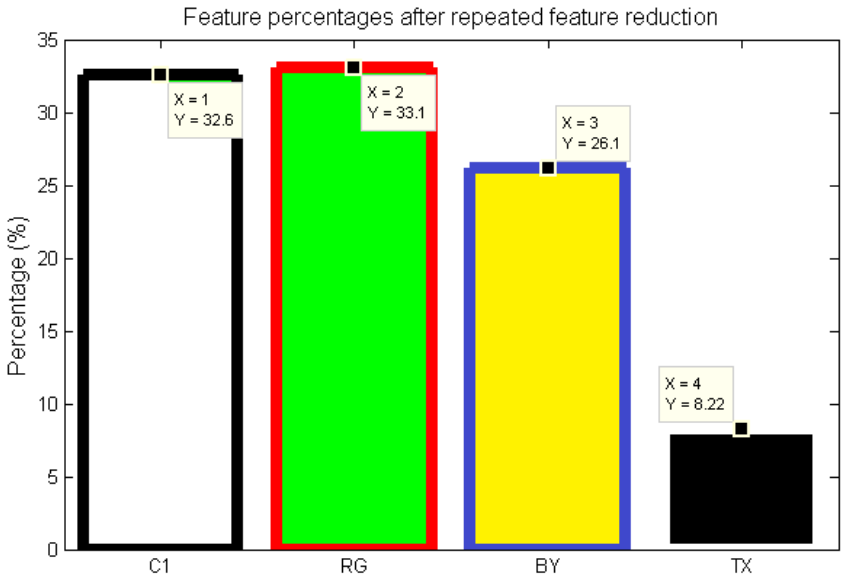
The total number of images for each class in dataset TX10 is 40 and so, this restricted the use of large featurebooks for experimentation in the rest of the datasets. Nevertheless, given the computational complexity of the Multiple channel and Gabor statistical parameterisation methods within the MATLAB environment, it was convenient to set the training sample size to 20 images per category and the testing size also to 20. The 50 features per image was also followed here, so the total number of features for each dataset varies according to the number of classes, e.g. TX10 has 10 classes and with 20 images for training each, has a total number of 10000 features. With regards to classification, in sections 6.2 and 7.1.1, it was observed that Adaboost is a fast classifier which shows comparable results with the other classifiers used in this thesis, without any necessary parameterisation procedures. To ensure fast, enhanced and accurate representations of results it was also preferred in this section with 1000 iterations, 100 weaklearners (perceptrons), λ = 0.001 and ε = 1.

Two predecessor algorithms were tested also here for comparison reasons against the newly developed texture additions, the MFHLib and COR100. Two COR100 algorithms are used in Table 7-6 below. The COR100 named “original”

is the algorithm developed in section 6.2 and as before it is used with morphological salience only while the number 100 signifies a percentage of 100% for features retained (i.e. only duplicates removed). The second COR100 algorithm named as “Four pyramids” is the same algorithm as COR100 with a fourth pyramid identical to the shape pyramid (i.e. two colour pyramids *RG, BY* and two pyramids of the same Gabor parameters as in section 7.2.1). The purpose of this modification is for experimental reasons only and illustrates how classification accuracy behaves when the ratio of features is changed. The ratio of features now (choice is unchanged to random) shifts between four pyramids two for shape and two for colour (Figure 7-8), rather than three pyramids in total two for colour and one for shape as in the original COR100 (Figure 6-13 and Figure 6-14).



(a)



(b)

Figure 7-8: (a) The distribution of feature extraction percentages amongst the four pyramids C1 – shape, RG – Red/Green, BY – Blue/Yellow and TX – Texture, (b) The distribution of feature extraction percentages amongst the four pyramids after feature reduction. Distributions were taken from the "Birds" dataset, second run.

It is not surprising that in Figure 7-8 (b), the contributing percentage for "TX" pyramid in the featurebook after feature reduction has dropped to only 8.22%, since COR100 "four pyramids" has the parameters for "TX" pyramid exactly identical. The algorithm is robust enough to detect the duplicates and leave out new features from "TX". This brings the total percentage for shape in reality (the summation of shape and texture pyramids) to 40.8%.

The texture feature methods were developed from the SFHLib and COR100 algorithms (section 6.2). Texture additions carry the added "-TX" suffix and are presented in Table 7-6. Methodologies are presented according to order of appearance in this section.

Dataset / Method	TX10*	CUUD	Butterflies	Birds	Cats	10 class
MFHLIB	37.3	62.5	40.5	40.8	43.6	36.3
COR100 - Original	46.7	67.1	53.6	49.7	50	54.2
COR100 - (Four pyramids)	45.1	70	56.1	52.5	43.6	53
CORTX100 – Hardcoded	24.5	56.7	54.7	53.9	31.7	36.2
CORTX100 – Multiple Channel	44	74.2	52.2	52.8	48.3	53.3
CORTX100 – Max Response	45.5	74.6	54.2	48	45.5	52.5
CORTX100 – Statistical Gabor	45.7	72.6	51.4	48.6	48.1	56.5

Table 7-6: The classification accuracies over 3 independent runs for texture recognition algorithms. All results typically vary at ±1.5. *TX10 is a greyscale image database therefore the use of Colour has no effect on the performance.

At first glance, it is obvious that the original MFHLib behaved worse compared to COR100 from the previous sections. This highlights the crucial role colour has within the overall object recognition process. The difference between the feature ratios in COR100 original and COR100 four pyramids is also evident.
The worst overall results were produced from the Hardcoded method. Its simplistic nature and fixed set of Gabor parameters as expected cannot generalise well over all images in the datasets and it seems that this prohibits distinct object description. For the two datasets "Butterflies" and "Birds" it is considered coincidental and dataset-dependent that the Hardcoded method improved the recognition performance significantly, compared to the rest of the methods. More importantly, the Hardcoded method highlights how critical the choice of Gabor parameters is within the overall process and justifies the development of the subsequent methodologies. The CORTX100 Multiple Channel technique in Table 7-6 demonstrates an overall substantially enhanced set of results across all datasets with respect to the Hardcoded method and marginally with respect to the other methods. It remains consistent as all algorithms with the addition of colour and outperforms MFHLib by a maximum of 17% for the "10 class" dataset and minimum of 4.7% for the "Cats" dataset. The Multiple Channel technique scored the highest classification accuracy percentage for the "Cats" dataset (excluding the original COR100 algorithm which cannot directly relate due to the different feature ratios). Importantly, the

Multiple Channel method exhibits improved overall scores against the COR100 four pyramids.
The CORTX100 Maximum Response generated overall the second worse set from the methods in this section. It produced the highest score for the CUUD dataset which was enhanced over the COR100 four pyramids and showed a similar trend for the rest of datasets. As with the Hardcoded Method, generalising the peak value of a Gabor histogram from the results here does not appear to increase performance more than the Multiple Channel approach.
The CORTX100 Statistical Gabor method produced an overall second best set of results only marginally behind the Multiple Channel. It showed the best performance on two datasets, the “TX10” and the “10 class”. Similarly to the other CORTX methods, it shows better behaviour when compared with the MFHLib and COR100 four pyramids algorithms.

7.2.6 Section Conclusions

The fluctuations in performance when tweaking Gabor parameters alone can highlight how the textural information affects the recognition process, which was not examined in depth in past literature. The novelty of this section was to present new methods for Gabor parameterisation and stress some of their problems for future research through their direct comparison within a biologically-inspired framework for machine vision.
Four novel Gabor parameterisation methodologies were presented in order to characterise the textural information of objects in the most optimum way for biologically-inspired object recognition. The textural information was treated in all cases as a separate feature to shape (general morphological information) and colour.
Generally, the Hardcoded technique produced the worst results. Nevertheless, for two datasets (Butterflies, Birds) it showed the best classification accuracies than any other method, mainly due to coincidental tuning of its parameters to these datasets. Its lack of optimisation and adaptability made the development of more sophisticated methods imperative. Conversely, the highest set of results was noticed with the Multiple Channel method. The improvement was noticeable by almost 17% for the 10 class dataset and a 4.7% minimum for the “Cats” dataset, when compared with MFHLib and even though, as mentioned in section 7.2.2, Gabor filter parameter σ (for computational reasons) is not varied. In the future experiments, on a spectrum of σ values and N template sizes would more precisely measure the efficiency of this biologically-inspired method.
The Maximum Response and Statistical Gabor parameterisation techniques produced comparable results, with the Statistical Gabor parameterisation being the second best. They both showed promising results, which were overall,

better than that of MFHLib and COR100 (four pyramids). It is considered a drawback here that one set of Gabor parameters characterises the whole scene and so, both methods would benefit in future experiments from a window approach, i.e. where their evaluation does not apply to the whole image but in areas of saliency independently. More experiments with a higher number of images and features across more datasets are also necessary to fully validate the proposed improvements.

Another key element to note for future experimentation is that the Gabor filter parameterisation described in this section is for object recognition only and thus Gabor filters in visual salience were unexamined. In the future, experimental analysis on the effect Gabor parameterisation has in visual attention would perhaps establish an additional texture feature (in addition i.e. to shape, colour, intensity, motion etc) for visual attention.

8 CONCLUSIONS

For the first time in literature a unified bottom-up saliency and recognition model was introduced with as many biological vision characteristics. Major contributions of this thesis were:

- A salient feature-based method for semantic feature extraction.
- Introduction of new image databases. In-depth experimentation of the algorithms with a greater variety of databases and classifiers.
- The design and integration of colour features coupled with the existing morphological-based features for a significantly improved biologically-inspired object recognition.
- A new illumination invariance property using colour constancy under a biologically-inspired framework.
- Rotation invariance methods that improve robustness over the original models.
- Novel adaptive Gabor filters for accurate texture information, enhancing the morphological description of objects and improving the overall classification performance.

This thesis more specifically proposed a series of design improvements over the existing biologically-inspired models of visual attention and object recognition. It has been shown that incorporating visual saliency into the ventral stream process is at minimum a two step process for bottom-up cases. The first step involves the detection of proto-objects around the scene. Under this step, important issues were identified concerning the precise number of images versus number of features that best describe objects without prior knowledge.

Another important issue addressed in the visual streams cooperation, is the rotation invariance of features. It is very often in practical terms that camera equipment or objects are not aligned at the same angle for every image taken and a rotation invariance property addresses this problem. Rotation invariance experiments under the developed methods here have shown inconclusive recognition performance when tested at various angles. However, with the promising results obtained in some cases, further investigation and experimentation would validate the existing and other future biologically-inspired rotation-invariant mechanisms stemming from the work here.

As a second step in the visual attention biologically-inspired process and by using prominent features within the ROI of the first step, semantic object information can be extracted without having to explicitly analyse image areas pixel-by-pixel or scanning the whole image with a sliding window approach. The proposed method has shown better recognition performance with respect to the

random and aimless manner by which features were being extracted from previous models. A structured approach to feature extraction paved the way for the addition of layers in the recognition part of the algorithm. With these new layers discarding redundant information and refining the total feature numbers the method has improved the integrity of the stored feature library without sacrificing performance.

The spectral information of images prior to this work had not been utilised systematically or tested thoroughly. Using biologically-inspired mechanisms this rich source of information was exploited and enhanced recognition performance was validated across various datasets. Moreover, it was essential to couple the newly adopted colour feature technique effectively with morphological information and establish a new relationship with salience. Experiments using this algorithm were conducted with two salience methods morphological-only versus morphological plus spectral salience. The latter showed better performance overall but results indicate a more dataset-dependent relationship between them.

With the introduced colour features a major question arises concerning their accurate spectral perception. Illumination invariant mechanisms such as colour constancy methods have been specifically developed to remedy the large fluctuation of illumination values found from image to image thus aiding towards generalised spectral cognition characteristics. There is an abundance of colour constancy methods and one can almost never guarantee a colour constancy method will be fully adaptive or the best performing. Previous work has shown that an adaptive colour constant mechanism should identify the best approach and such a methodology is tested here. Results show the need for further experimentation as the comparative analysis between colour constancy methods and their fusion did not enhance classification accuracies.

Four different methods for Gabor filter parameterisation and design were investigated. For the first time, a generic method of morphological representation is implemented. Using the histogram characteristics of the Gabor filter response along with Weibull distribution matching, the scene's characteristics point to the appropriate set of Gabor parameters in order to capture texture information. This information is used in conjunction with the existing shape and spectral feature methods of the algorithm and has shown an improved performance. The comparison between methods has highlighted the importance of an elaborate Gabor mechanism since without its existence, tuning across all visual scenes may lead to poor performance.

8.1 Future work

Future work on rotation invariance requires additional computational power. Future experiments on the rotation invariant algorithms should expand with more datasets, different parameterisations and scenarios. Additional rotation invariance methodologies should be also implemented and employed based on future physiological data which will provide more evidence on the biological process itself.

For saliency it is planned to use extensive and dynamic visual scenarios to verify the developed algorithms for adaptability. Further experimentation for the COR methodologies would indicate whether choosing morphology only in visual attention is better than a morphology and spectral visual attention approach. Perhaps with further experimentation i.e. a greater variety of datasets, parameters and scenarios, a predictable dataset-dependent behaviour can emerge. More work is also necessary for an adaptive class-by-class feature reduction approach on the S3/C3 layers.

Additionally, there is need for an improved adaptive illumination mechanism, an additional photometric analysis and the creation of an object recognition dataset with ground truth data. Further experimentation on the illumination invariance mechanism should also expand on an extensive colour constancy dataset that encompasses a greater variety of scenes (i.e. city environments, indoor images etc), and preferably one that is also used by the object recognition mechanism. In addition, a greater number of colour constancy methods should be examined and alternative biologically-inspired illumination invariant mechanisms should also be considered.

A fully adaptive biologically-inspired Gabor parameterisation is imperative in order to consistently improve the integrity of the classification performance especially in different texture databases. An experimental analysis on the effect it has over the visual attention process would perhaps also establish an additional texture feature (in addition i.e. to shape, colour, intensity, motion etc) for visual attention.

Future work should also involve the expansion of the visual attention model (CUVS) in order to accommodate extra features such as size and depth. The stereoscopic depth feature would enable efficient distinction between foreground and background objects. In addition, biologically-inspired stereoscopic features are expected to enhance performance and biological realism.

Significant contributions can originate from successfully applying the biologically-inspired techniques in spiking neural networks. The learning process could be enhanced by using biological-like methods such as reinforcement learning and/or Hebbian-based learning. Spiking neural networks can furthermore lead to the utilisation of Graphics Processing Unit (GPU) processors for increased speed/simulation and ultimately for the development of neuromorphic architectures that mimic the visual techniques more realistically.

REFERENCES

[1] M. E. Rosheim, *Leonardo's lost robots*. Springer Science & Business, 2006.

[2] S. Y. Nof, *Handbook of Industrial Robotics*. John Wiley & Sons, 1999.

[3] I. Spectrum, "MoNETA: A Mind Made from Memristors." [Online]. Available: http://spectrum.ieee.org/robotics/artificial-intel.

[4] MRC, "Neurogrid." [Online]. Available: http://www.neurogrid.ac.uk/.

[5] EPFL, "THE BLUE BRAIN PROJECT EPFL." [Online]. Available: http://bluebrain.epfl.ch/.

[6] "Neuroscience Gallery," 2009. [Online]. Available: http://www.conncad.com/gallery/single_neurons.html.

[7] I. Wikipedia, "Neuron." [Online]. Available: http://en.wikipedia.org/wiki/Neuron.

[8] G. M. Shepherd, *The Synaptic Organization of the Brain*. USA: Oxford University Press, 2004.

[9] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of Neural Science*. USA: McGraw-Hill, 2000, p. 1414.

[10] W. Lytton W, *From Computer to Brain, Foundations of Computational Neuroscience*. New York, USA: Springer, 2002.

[11] W. Rall, "Branching dendritic trees and motoneuron membrane resistivity," *Experimental Neurology*, vol. 1, no. 5, pp. 491–527, 1959.

[12] H. Hering and M. Sheng, "Dendritic Spines: Structure, Dynamics and Regulation," *MacMillan Magazines*, vol. 2, pp. 880–888, 2001.

[13] G. Yang, F. Pan, and W.-B. Gan, "Stably maintained dendritic spines are associated with lifelong memories," *Nature*, vol. 462, pp. 920–924, 2009.

[14] W. Maas, A. M. Zador, and C. M. Bishop, "Computing and Learning with Dynamic Synapses," in *Pulsed Neural Networks*, Cambridge, MA, USA: MIT press, 1999, pp. 321–336.

[15] H. McBride M, M. Neuspiel, and S. Wasiak, "Mitochondria: more than just a powerhouse.," *Current Biology*, vol. 16, no. 14, pp. 551–560, 2006.

[16] A. Hodgkin and A. Huxley, " A quantitative description of membrane current and its application to conduction and excitation in nerve," *Journal of Physiology*, vol. 117, pp. 500–544, 1952.

[17] L. R. Squire, F. E. Bloom, and N. C. Spitzer, *Fundamental Neuroscience*. London, UK: Elsevier, 2008.

[18] W. Inc, "Hyperpolarization." [Online]. Available: http://en.wikipedia.org/wiki/Hyperpolarization_(biology).

[19] A. Bain, *Mind and Body: The Theories of Their Relation*. New York, USA: D. Appleton and Company, 1873.

[20] W. McCullouch and W. Pitts, "A Logical Calculus of Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[21] D. Hebb, *The Organization of Behavior*. New York, USA: Wiley, 1949.

[22] L. B. Abbott, "Lapicque's introduction of the integrate-and-fire model neuron (1907)," *Brain Research Bulletin*, vol. 50, pp. 303–304, 1999.

[23] W. Gerstner and W. Kistler, *Spiking Neuron Models. Single Neurons, Populations, Plasticity.* Cambridge university press, 2002.

[24] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," *Biophysical*, vol. 1, pp. 445–466, 1961.

[25] C. Morris and H. Lecar, " Voltage Oscillations in the barnacle giant muscle fiber," *Biophysical*, vol. 35, pp. 193–213, 1981.

[26] H. Hindmarsh and R. M. Rose, "A model of neuronal bursting using three coupled first-order differential equations," in *Proceedings of the Royal Society* , 1984, pp. 87–102.

[27] W. Maas and A. M. Zador, "Dynamic Stochastic Synapses as Computational Units," *Neural Computation*, vol. 11, pp. 903–917, 1999.

[28] H. Mikram, Y. Wang, and M. Tsodyks, "Differential signaling via the same axon of neocortical pyramidal neurons," *Proceedings of the National Academy of Science USA*, vol. 95, pp. 5323–5328, 1998.

[29] E. D. Andrian and Y. Zotterman, "The impulses produced by sensory nerve-endings Part II. The response of a Single End-Organ," *Journal of Physiology*, vol. 61, no. 2, pp. 151–171, 1926.

[30] L. T. Troland, "The psychophysiology of auditory qualities and attributes.," *Journal of General Psychology*, vol. 2, pp. 28–58, 1929.

[31] A. P. Georgopoulos, R. Kettner, and A. B. Schwartz, “Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population,” *Journal of Neuroscience*, vol. 8, pp. 2928–2937, 1988.

[32] R. R. de Ruyter van Steveninck, G. D. Lewen, S. P. Strong, R. Koberle, and W. Bialek, “Reproducibility and variability in neural spike trains.,” *Science*, vol. 275, no. 5307, pp. 1805–1808, 1997.

[33] M. J. Berry, D. K. Warland, and M. Meister, “The structure and precision of retinal spike trains,” in *Proceedings of the National Academy of Science*, 1997, pp. 5411–5416.

[34] R. VanRullen and S. J. Thorpe, “Rate Coding vs Temporal Order Coding: What the Retinal Ganglion Cells tell the Visual Cortex.,” *Neural Computation*, vol. 13, no. 6, pp. 1255–1283, 2001.

[35] M. A. Arbib, *The Handbook of Brain Theory and Neural Networks*. MIT Press, 2002.

[36] C. Glackin, L. McDaid, L. Maguire, and H. Sayers, “Implementing Fuzzy Reasoning on a Spiking Neural Network,” in *ICANN* , 2008, pp. 258–267.

[37] S. Bohte, J. Kok, and H. L. Poutre, “Spike-prop: error-backprogation in multi-layer networks of spiking neurons,” in *Proceedings of the European Symposium on Artificial Neural Networks ESANN*, 2000, pp. 419–425.

[38] P. A. Tino and J. S. Mills, “Learning Beyond Finite Memory in Recurrent Networks of Spiking Neurons.,” *Advances in Natural Computation*, pp. 666–675, 2005.

[39] W. Maas, T. Natschlager, and H. Makram, “Real-Time ComputingWithout Stable States:A New Framework for Neural Computation Based on Perturbations,” *Neural Computation*, vol. 14, pp. 2531–2560, 2002.

[40] M. O’Halloran, S. Cawley, B. McGinley, R. Conceicao, F. Morgan, E. Jones, and M. Glavin, “Evolving spiking neural network topologies for breast cancer classification in a dielectrically hetereogeneous breast,” *Progress In Electromagnetics Research Letters*, vol. 25, pp. 153–162, 2011.

[41] H. N. A. Hamed, N. Kasabov, Z. Michlovsky, and S. S. M, “String Pattern Recognition Using Evolving Spiking Neural Networks and Quantum Inspired Particle Swarm Optimization,” in *16th International Conference on Neural Information Processing*, 2009, pp. 611–619.

[42] A. Upegui, A. Pena-Reyes, and E. Sanchez, “A methodology for evolving spiking neural-network topologies on-line using partial dynamic

configuration.," in *Proceedings of ICCI - International Conference on Computational Inteligence*, 2003.

[43] S. G. Wysoski, L. Benuskova, and N. Kasabov, "Fast and adaptive network of spiking neurons for multi-view visual pattern recognition," *Neurocomputing*, vol. 71, pp. 2563–2575, 2008.

[44] S. G. Wysoski, L. Benuskova, and N. Kasabov, "Evolving spiking neural networks for audiovisual information processing," *Neural Networks*, vol. 23, no. 7, pp. 819–835, 2010.

[45] W. Maas, T. Natschlaeger, and H. Markram, "A model for real-time computation in generic neural microcircuits.," *Advances in Neural Information Processing Systems, NIPS*, vol. 15, pp. 229–236, 2003.

[46] T. Yamazaki and S. Tanaka, "The cerebellum as a liquid state machine.," *International Neural Network Society*, vol. 20, no. 3, pp. 290–297, 2007.

[47] B. Jones, D. Stekelo, J. Rowe, and C. Fernando, "Is there a Liquid State Machine in the Bacterium Escherichia Coli?," in *2007 IEEE Symposium on Artificial Life*, 2007, pp. 187–191.

[48] S. Kok, "Liquid State Machine Optimization," Utrecht, 2007.

[49] M. Lukosevicius and H. Jaeger, "Reservoir Computing Approaches to Recurrent Neural Network Training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[50] H. Markram, J. Lubke, M. Frotscher, and B. Sakmann, " Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs," *Science*, vol. 275, no. 5297, pp. 213–215, 1997.

[51] G. Q. Bi and M. M. Poo, "Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Post-synaptic Cell Type," *The Journal of Neuroscience*, vol. 18, no. 24, pp. 10464–10472, 1998.

[52] R. Legenstein, C. Naeger, and W. Maass, "What can a Neuron Learn with Spike-Timing-Dependent Plasticity?," *Neural Computation*, vol. 17, no. 11, pp. 2337–2382, 2005.

[53] F. Ponulak and A. Kasinski, "Supervised Learning in Spiking Neural Networks with ReSuMe: Sequence Learning, Classification and Spike-Shifting," *Neural Computation*, vol. 22, no. 2, pp. 467–510, 2009.

[54] "Pass my exams." [Online]. Available: http://www.passmyexams.co.uk/GCSE/physics/use-of-lenses-for-correcting-vision-eyesight.html.

[55] L. M. Chalupa and J. S. Werner, *The Visual Neurosciences Volume*. London, Cambridge, MA: MIT Press, 2004.

[56] D. H. Hubel, *Eye, Brain and Vision*. New York, USA: Scientific American Library, 1988.

[57] B. E. Goldstein, *Sensation and Perception*. Belmont, USA: Wadsworth, 2010.

[58] I. Provencio, I. R. Rodriguez, G. Jiang, W. P. Hayes, E. F. Moreira, and M. D. Rollag, "A Novel Human Opsin in the Inner Retina," *The Journal of Neuroscience*, vol. 20, no. 2, pp. 600–605, 2000.

[59] J. K. Bowmaker and H. J. Dartnall, "Visual pigments of rods and cones in a human retina.," *Journal of Physiology*, vol. 298, pp. 501–511, 1980.

[60] C. Curcio A and K. Allen A, "Topography of ganglion cells in human retina," *J Comp Neurol.*, vol. 312, no. 4, pp. 610–624, 1991.

[61] D. H. Hubel and T. N. Wiesel, *Brain and Visual Perception: the story of a 25-year collaboration*. New York: Oxford University Press, 2005.

[62] C. Enroth-Cugell and J. G. Robson, "The Contrast Sensitivity of Retinal Ganglion Cells of the Cat.," *Journal of Physiology*, vol. 187, pp. 517–523, 1966.

[63] S. Engel, X. Zhang, and B. Wandell, "Colour Tuning in Human Visual Cortex Measured with functional Magnetic Resonance Imaging," *Nature*, vol. 388, no. 6637, pp. 68–71, 1997.

[64] D. H. Brainard and B. A. Wandell, "Analysis of the retinex theoy of color vision," *Journal of Optical Society of America*, vol. 3, no. 10, pp. 1651–1660, 1986.

[65] B. R. Conway and M. S. Livingstone, "Spatial and Temporal Properties of Cone Signals in Alert Macaque Primary Visual Cortex," *The Journal of Neuroscience*, vol. 26, no. 42, pp. 10826–10846, 2006.

[66] S. D. Hordley, "Scene Illuminant Estimation: Past, Present and Future," *Color Research & Application*, vol. 31, no. 4, pp. 303–314, 2006.

[67] E. H. Land, "The Retinex Theory of Color Constancy," *Scientific American*, pp. 108–129, 1977.

[68] E. Land and J. McCann, "Lightness and retinex theory," *The Journal of the Optical Society of America A.*, vol. 61, no. 1, pp. 1–11, 1971.

[69] M. Ebner, *Color constancy*. Wiley and Sons, 2007.

[70] V. Agrawal, “An Overview of Color Constancy Algorithms,” *Journal of Pattern Recognition Research*, vol. 1, pp. 42–54, 2006.

[71] R. Gershon, A. D. Jepson, and J. K. Tsotsos, “From [r,g,b] to Surface Reflectance: Computing Color Constant Descriptors in Images,” *Perception*, pp. 755–758, 1988.

[72] L. T. Maloney and B. A. Wandell, “Color Constancy: A Method for Recovering Surface Reflectance,” *Journal of Optical Society of America A*, vol. 3, no. 1, pp. 29–33, 1986.

[73] B. K. P. Horn, “Determining Lightness from an Image,” *Computer Vision, Graphics and Image Processing*, vol. 3, pp. 277–299, 1974.

[74] G. D. Forsyth, “A Novel Algorithm for Color Constancy,” *International Journal of Computer Vision*, vol. 5, no. 1, pp. 5–36, 1990.

[75] J. A. Worthey and M. H. Brill, “Heuristic Analysis of Von Kries Color Constancy,” *Journal of Optical Society of America A*, vol. 3, pp. 1708–1712, 1986.

[76] C. Rosenberg, T. Minka, and A. Ladsariya, “Bayesian Color Constancy with Non-Gaussian,” *NIPS*, 2003.

[77] G. Buchsbaum, “A Spatial Processor Model for Object Color Perception,” *Journal of Franklin Institute*, vol. 310, pp. 1–26, 1980.

[78] K. Brainard, “A Comparison of Computational Color Constancy Algorithms - Part I: Methodology and Experiments with Synthesized Data,” *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 972–983, 2002.

[79] K. Barnard and B. V. Funt, “Investigation into Multiscale Retinex,” *Color Imaging in Multimedia*, pp. 9–17, 1998.

[80] B. V. Funt, K. Barnard, and K. Brockington, “Luminance Based Multiscale Retinex,” *AIC International Color Association*, 1997.

[81] A. Rizzi, C. Gatta, and D. Marini, “From Retinex to Automatic Color Equalization: Issues in Developing a New Algorithm for Unsupervised Color Equalization,” *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 75–84, 2004.

[82] D. J. Jobson and G. A. Woodell, “Properties of a Center/Surround Retinex Part 2 - Surround Design,” Hampton, Virginia, 1995.

[83] D. J. Jobson, Z. Rahman, and G. A. Woodell, “A Multiscale Retinex for Bridging the Gap Between Color Images and the Human Observation of

Scenes," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 965–976, 1997.

[84] G. D. Finlayson, S. D. Hordley, and R. Xu, "Convex programming colour constancy with a diagonal-offset model," *IEEE International Conference on Image processing*, pp. 51–948, 2005.

[85] C. Rosenberg, M. Hebert, and S. Thrun, "Color Constancy Using KL-Divergence," in *Proceedings of International Conference on Computer Vision*, 2001, pp. 239–246.

[86] W. T. Freeman and D. H. Brainard, "Bayesian Decision Theory, the Maximum Local Mass Estimate, and Color Constancy," in *Proceedings of IEEE 5th International Conference on Computer Vision*, 1995, pp. 210–217.

[87] G. D. Finlayson, "Coefficient of Color Constancy," 1995.

[88] G. H. Finlayson, S. D. Hordley, and P. M. Hubel, "Color by Correlation: A Simple, Unifying Framework for Color Constancy," *EEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1209–1221, 2001.

[89] V. C. Cardei and B. V. Funt, "Committee-based color constancy," in *Proceedings IS and T/SID Color Imaging*, 1999, pp. 311–313.

[90] B. V. Funt and V. Cardei, "Bootstrapping Color Constancy," *Proceedings of SPIE, Electronic Imaging IV*, vol. 3644, 1999.

[91] M. Ebner, "A Parallel Algorithm for Color Constancy," *Journal of Parallel and Distributed Computing*, vol. 64, no. 1, pp. 79–88, 2004.

[92] A. Moore, J. Allman, and R. M. Goodman, "Real Time Neural System for Color Constancy," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 237–247, 1991.

[93] R. Stanikunas, H. Vaitkevicius, and J. J. Kulikowski, "Investigation of Color constancy with a Neural Network," *Neural Networks*, vol. 17, pp. 327–337, 2004.

[94] W. C. Huang and C. H. Wu, "Adaptive Color Image Processing and Recognition for Varying Backgrounds and Illumination Conditions," *IEEE Transactions on Industrial Electronics*, vol. 45, no. 2, pp. 351–357, 1998.

[95] B. Funt and W. Xiong, "Estimating Illumination Chromaticity via Support Vector Regression," in *Proceedings of 12th Color Imaging Conference: Color Science and Engineering Systems and Applications*, 2004, pp. 47–52.

[96] A. Gijsenij and T. Gevers, “Color Constancy using Natural Image Statistics and Scene Semantics,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, pp. 687–698, 2011.

[97] A. Oliva and A. Torralba, “Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope,” *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[98] A. Torralba and A. Oliva, “Statistics of natural image categories,” *Network: Computation in Neural Systems*, vol. 14, pp. 391–412, 2003.

[99] J. M. Geusebroek and A. W. M. Smeulders, “A Six-Stimulus Theory for Stochastic Texture,” *International Journal of Computer Vision*, vol. 62, no. 1/2, pp. 7–16, 2005.

[100] B. Lee B and H. Sun, “The chromatic input to cells of the magnocellular pathway of primates,” *Journal of Vision*, vol. 9, no. 2, pp. 1–18, 2009.

[101] E. M. Blessing, S. G. Solomon, M. Hashemi-Nezhad, B. J. Morris, and P. R. Martin, “Chromatic and spatial properties of parvocellular cells in the lateral geniculate nucleus of the marmoset (Callithrix jacchus),” *Journal Physiology*, vol. 557, no. 1, pp. 229–245, 2004.

[102] A. White J, R, S. Solomon G, and P. Martin R, “Spatial properties of koniocellular cells in the lateral geniculate nucleus of the marmoset (Callithrix jacchus),” *Journal Physiology*, vol. 533, no. 2, pp. 519–535, 2001.

[103] J. P. Van Kleef, S. L. Cloherty, and M. R. Ibbotson, “Complex cell receptive fields: evidence for a hierarchical mechanism,” *Journal of Physiology*, vol. 588, no. 18, pp. 3457–3470, 2010.

[104] C. R. Michael, “Color-Sensitive Hypercomplex Cells in Monkey Striate Cortex,” *Journal of Neurophysiology*, vol. 42, no. 3, 1979.

[105] J. G. Daugman, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters,” *Journal of Optical Society of America*, vol. 2, no. 7, pp. 1160–1169, 1985.

[106] J. P. Jones and L. A. Palmer, “An Evaluation of the Two- Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex,” *Journal of Neurophysiology*, vol. 58, no. 6, pp. 1233–1258, 1987.

[107] D. Cope, B. Blakeslee, and M. E. McCourt, “Simple cell response properties imply receptive field structure: balanced Gabor and/or bandlimited field functions,” *Journal of Optical Society of America A*, vol. 26, no. 9, pp. 2067–2092, 2009.

[108] N. Petkov and P. Kruizinga, “Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells,” *Biological cybernetics*, vol. 76, pp. 83–96, 1997.

[109] R. DeValois, D. Albrecht, and L. Thorell, “Spatial Frequency Selectivity of Cells in Macaque Visual Cortex,” *Vision Research*, vol. 22, pp. 545–559, 1982.

[110] J. J. Koenderink and A. J. Van Doorn, “Representation of local geometry in the visual system,” *Biological cybernetics*, vol. 55, no. 6, pp. 367–375, 1987.

[111] R. A. Young, R. M. Lesperance, and W. W. Meyer, “The Gaussian Derivative model for spatial-temporal vision: I. Cortical model,” *Spatial Vision*, vol. 3, no. 4, pp. 261–319, 2001.

[112] I. M. Finn and D. Ferster, “Computational Diversity in Complex Cells of Cat Primary Visual Cortex,” *The Journal of Neuroscience*, vol. 27, no. 36, pp. 9638–9648, 2007.

[113] L. Shams and C. von der Malsburg, “The role of complex cells in object recognition,” *Vision Research*, vol. 42, pp. 2547–2554, 2002.

[114] H. Spitzer and S. Hochstein, “A complex-cell receptive-field model.,” *Journal of Neurophysiology*, vol. 53, no. 5, pp. 1266–1286, 1985.

[115] M. Hansard and R. Horaund, “A Differential Model of the Complex Cell,” *Journal of Neural Computation*, vol. 23, no. 9, pp. 2324–2357, 2011.

[116] D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” *Journal of Physiology*, vol. 195, no. 1, pp. 215–243., 1967.

[117] M. Riesenhuber and T. Poggio, “Hierarchical models of object recognition in cortex,” *Nat. Neurosci.*, no. 2(11):1019–25, 1999.

[118] H. H. Bulthoff and S. Edelman, “Psychophysical support for a two-dimensional view interpolation theory of object recognition,” *Proceedings of the National Academy of Science USA*, vol. 89, pp. 60–64, 1992.

[119] M. C. Booth and E. T. Rolls, “View-invariant representations of familiar objects by neurons in the inferior temporal visual cortex,” *Cerebral Cortex*, vol. 8, no. 6, pp. 510–523, 1998.

[120] N. K. Logothetis, J. Pauls, and T. Poggio, “Shape representation in the inferior temporal cortex of monkeys,” *Current Biology*, vol. 5, pp. 552–563, 1995.

[121] A. L. Yarbus, “Eye movements and vision,” *Science*, vol. 160, no. 3828, p. 657, 1967.

[122] R. L. Didday and M. A. Arbib, “Eye movements and Visual Perception,” *International Journal Man-Machine studies*, vol. 7, no. 4, pp. 547–569, 1975.

[123] G. Ungerleider L and M. Mishkin, “Two cortical visual systems,” no. In Analysis of visual behavior (ed. D. J. Ingle, M. A. Goodale & R. J. W. Mansfield), 1982.

[124] M. Mishkin, G. Ungerleider, and A. K. Macko, “Object vision and spatial vision: Two cortical pathways,” *Trends in neuroscience*, vol. 6, pp. 414–417, 1983.

[125] I. Wikipedia, “Broadmann Areas.” [Online]. Available: http://en.wikipedia.org/wiki/Brodmann_area.

[126] C. I. of H. Research, “The brain.” [Online]. Available: http://thebrain.mcgill.ca.

[127] F. Qiu T and R. Heydt, “Figure and Ground in the Visual Cortex: V2 Combines Sterescopic cues with Gestalt Rules,” *Neuron*, vol. 47, pp. 155–166, 2005.

[128] M. F. López-Aranda, J. F. López-Téllez, I. Navarro-Lobato, M. Masmudi-Martín, A. Gutiérrez, and Z. U. Khan, “Role of Layer 6 of V2 Visual Cortex in Object-Recognition Memory,” *Science*, vol. 325, pp. 87–89, 2009.

[129] S. Zeki, “Improbable areas in the visual brain,” *Elsevier*, vol. 26, no. 1, pp. 23–26, 2003.

[130] K. Gegenfurtner R, D. Kiper C, and J. Levitt B, “Functional Properties of Neurons in Macaque Area V3,” *Journal of Neurophysiology*, vol. 77, no. 4, pp. 1906–1923, 1997.

[131] S. V. David, B. Y. Hayden, J. A. Mazer, and J. L. Gallant, “Attention to stimulus features shifts spectral tuning of V4 neurons during natural vision.,” *Neuron*, vol. 59, no. 3, pp. 509–521, 2008.

[132] D. Felleman and V. Essen D, “Distributed hierarchical processing in the primate cerebral cortex,” *Cerebral Cortex*, vol. 1, no. 1, pp. 1–47, 1991.

[133] A. J. Movshon, H. E. Adelson, S. M. Gizzi, and T. W. Newsome, “The analysis of moving visual patterns,” *C. Chagas, R. Gattass, & C. Gross (Eds.), Pattern recognition mechanisms*, pp. 117–151, 1985.

[134] S. Zeki, “Thirty years of a very special visual area, Area V5,” *Journal Pshysiology*, vol. 557, no. 1, pp. 1–2, 2004.

[135] C. Galetti, P. Fattori, M. Gamberini, and D. F. Kutz, “The cortical visual area V6: brain location and visual topography,” *The European Journal of Neuroscience*, vol. 11, no. 11, pp. 3922–3936, 1999.

[136] P. Fattori, S. Pitzalis, and C. Galetti, “The cortical visual area V6 in macaque and human brains,” *Journal of Physiology, Paris*, vol. 103, no. 1–2, pp. 88–97, 2009.

[137] Y. Sasaki, W. Vanduffel, T. Knutsen, C. Tyler, and R. Tootell, “Symmetry activates extrastriate visual cortex in human and nonhuman primates,” in *Proceedings of National Academic Sciences (USA)*, 2005, pp. 3159–3163.

[138] H. Bridge and A. Parker J, “Topographical representation of binocular depth in the human visual cortex using fMRI,” *Journal of Vision*, vol. 7, no. 14, pp. 1–14, 2007.

[139] M. Carrasco, “Visual attention: The past 25 years,” *Vision Research*, vol. 51, pp. 1484–1525, 2011.

[140] S. Baldassi and P. Verghese, “Attention to locations and features: Different top-down modulation of detector weights.,” *Journal of Vision*, vol. 5, no. 6, pp. 556–570, 2005.

[141] S. Ling, T. Liu, and M. Carrasco, “How spatial and feature-based attention affect the gain and tuning of population responses.,” *Vision Research*, vol. 49, no. 10, pp. 1194–1204, 2009.

[142] T. Liu, S. T. Stevens, and M. Carrasco, “Comparing the time course and efficacy of spatial and feature-based attention.,” *Vision Research*, vol. 47, no. 1, pp. 108–113, 2007.

[143] J. H. Maunsell and S. Treue, “Feature-based attention in visual cortex.,” *Trends in neuroscience*, vol. 29, no. 6, pp. 317–322, 2006.

[144] B. J. Scholl, “Objects and attention: The state of the art.,” *Cognition*, vol. 80, no. 1–2, pp. 1–46, 2001.

[145] S. Frintrop, E. Rome, and H. Christensen I, “Computational Visual Attention Systems and their Cognitive Foundations: A Survey,” *ACM Transactions on Applied Perception (TAP)*, vol. 7, no. 1, 2010.

[146] S. Grossberg, “Biological Competiton: Decision Rules, pattern formation and oscillations,” *PNAS*, vol. 77, pp. 2338–2342, 1980.

[147] S. Corchs and G. Deco, “A neurodynamical model for selective visual attention using oscillators.,” *Neural Networks*, vol. 14, no. 8, pp. 981–990, 2001.

[148] G. Deco, O. Pollatos, and J. Zihl, “The time course of selective visual attention: theory and experiments,” *Vision Research*, vol. 42, pp. 2925–2945, 2002.

[149] G. Deco and J. Zihl, “A neurodynamical model of visual attention: feedback enhancement of spatial resolution in a hierarchical system.,” *Computational Neuroscience*, vol. 10, no. 3, pp. 231–253, 2001.

[150] F. H. Hamker, “A dynamic model of how feature cues guide spatial attention,” *Vision Research*, vol. 44, pp. 501–521, 2004.

[151] F. H. Hamker and M. Zirnsak, “V4 receptive field dynamics as predicted by a systems-level model of visual attention using feedback from the frontal eye field,” *Neural Networks*, vol. 19, pp. 1371–1382, 2006.

[152] J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. Lai Davis, N., and F. Nuflo, “Modeling visual attention via selective tuning,” *Artificial Intelligence*, vol. 78, no. 1–2, pp. 507–545, 1995.

[153] A. Zaharescu, A. L. Rothenstein, and J. K. Tsotsos, “Towards a Biologically Plausible Active Visual Search Model.,” *Attention and Performance in Computational Vision: Second International Workshop*, 2004.

[154] A. L. Rothenstein, A. J. Rodriguez-Sanchez, E. Simine, and J. K. Tsotsos, “Visual Feature Binding within the Selective Tuning Attention Framework.,” *International Journal of Pattern Recognition and Artificial Intelligence - Special Issue on Brain, Vision and Artificial Intelligence*, pp. 861–881, 2008.

[155] M. A. Treisman and G. Gelade, “A feature integration theory of attention,” *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[156] L. Itti, C. Koch, and E. Niebur, “A model of saliency based visual attention for rapid scene analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[157] R. Milanese, S. Gil, and T. Pun, “Attentive mechanisms for dynamic and static scene analysis,” vol. 34, no. 8, 2428–2434, 1995.

[158] S. Baluja and A. D. Pomerleau, “Expectation-based selective attention for visual monitoring and control of a robot vehicle,” *Robotics and autonomous systems*, vol. 22, no. 3–4, pp. 329–344, 1997.

[159] A. Torralba, A. Oliva, S. M. Castelhano, and J. M. Henderson, “Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search,” *Psychological Review*, vol. 113, no. 4, pp. 766–786, 2006.

[160] N. D. B. Bruce and J. K. Tsotsos, “Saliency based on information maximization,” *Advances in Neural Information Processing Systems*, vol. 18, pp. 155–162, 2006.

[161] L. Zhang, M. H. Tong, and G. W. Cottrell, “Information attracts attention: A probabilistic account of the cross-race advantage in visual search,” 2007.

[162] W. Kienzle, F. A. Wichmann, B. Scholkopf, and M. Franz, “A nonparametric approach to bottom-up visual saliency,” *Advances in Neural Information Processing Systems*, pp. 1–8, 2006.

[163] J. Harel, C. Koch, and P. Perona, “Graph-Based Visual Saliency,” *Advances in neural information processing systems 19*, 2007.

[164] D. Gao and N. Vasconcelos, “Bottom-up saliency is a discriminant process,” in *IEEE International Conference on Computer Vision*, 2007.

[165] C. Koch and S. Ullman, “Shifts in selective visual attention: towards the underlying neural circuitry,” no. 4, 219–227, 1985.

[166] J. Morgan M, A. Adam, and D. Mollon J, “Dichromats detect colour-camouflaged objects that are not detected by trichromats,” *Proceedings of Biological Science*, vol. 248, no. 1323, pp. 291–295, 1992.

[167] J. Tovee M, *An introduction to the visual system*. Cambridge, UK: Cambridge University Press, 1996.

[168] P. Burt and E. H. Adelson, “The Laplacian Pyramid as a Compact Image code,” *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[169] M. Grundland and N. A. Dodgson, “Decolorize: Fast, Contrast Enhancing, Color to Grayscale Conversion,” *Pattern Recognition*, vol. 40, no. 11, pp. 2891–2896, 2006.

[170] R. Gonzalez C and R. Woods E, *Digital Image processing*. New Jersey, USA: Prentice Hall, 2002.

[171] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications*, 2nd Editio. London, UK: Springer-Verlag, 2008, p. 798.

[172] E. Peli, “Contrast in complex images,” *Journal of Optical Society*, vol. 7, no. 10, pp. 2032–2040, 1990.

[173] R. Rao P, N, “Hierarchical Bayesian Inference in Networks of Spiking Neurons,” *Advances in NIPS*, vol. 17, 2005.

[174] T. Lee S and D. Mumford, “Hierarchical Bayesian inference in the visual cortex,” *Journal of Optical Society of America*, vol. 20, no. 7, pp. 1434–1447, 2003.

[175] A. Wade R, A. Brewer A, and J. Rieger W, “Functional measurements of human ventral occipital cortex: retinotopy and colour,” *Transactions of the Royal Society*, vol. 357, pp. 963–973, 2002.

[176] N. Hadjikhani, A. Liu K, A. Dale M, P. Cavanagh, and R. Tootell B, “Retinotopy and color sensitivity in human visual cortical area V8,” *Nat. Neuroscience*, vol. 1, no. 3, pp. 235–241, 1998.

[177] R. Gatass, P. Sousa, and C. Gross G, “Visuotopic organization and extent of V3 and V4 of the macaque,” *Journal of Neuroscience*, vol. 8, pp. 1831–1845, 1988.

[178] M. Ito, H. Tamura, I. Fujita, and K. Tanaka, “Size and Position Invariance of Neuronal Responses in Monkey Inferotemporal Cortex,” *Journal of Neurophysiology*, vol. 73, no. 1, pp. 218–226, 1995.

[179] G. Sary, R. Vogels, and G. A. Orban, “Cue-invariant shape selectivity of macaque inferior temporal neurons,” *Science*, vol. 260, no. 5110, pp. 995–997, 1993.

[180] S. Brincat L and C. Connor E, “Dynamic Shape Synthesis in Posterior Inferotemporal Cortex,” *Elsevier*, vol. 49, no. 1, pp. 17–24, 2006.

[181] Y. Liu and B. Jagadeesh, “Neural Selectivity in Anterior Inferotemporal Cortex for Morphed Photographic Images During Behavioral Classification or Fixation,” *Journal Neurophysiology*, vol. 100, no. 2, pp. 966–982, 2007.

[182] N. C. Aggelopoulos and E. T. Rolls, “Scene perception: inferior temporal cortex neurons encode the positions of different objects in the scene,” *European Journal of Neuroscience*, vol. 22, pp. 2903–2916, 2005.

[183] J. J. DiCarlo and J. H. Maunsell, “Anterior inferotemporal neurons of monkeys engaged in object recognition can be highly sensitive to object retinal position.,” *Journal of Neurophysiology*, vol. 89, no. 6, pp. 3264–3278, 2003.

[184] T. Serre, L. Wolf, S. Bileschi, and M. Riesenhuber, “Robust Object Recognition with Cortex-like mechanisms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411–426, 2007.

[185] K. Fukushima, “Neocognitron: A self organizing neural network for a mechanism of pattern recognition unaffected by shift in position,” *Biological cybernetics*, vol. 36, no. 4, pp. 93–202, 1980.

[186] B. W. Mel, “SEEMORE: Combining color, shape and texture histogramming in a neurally-inspired approach to visual object recognition,” *Neural Computation*, vol. 9, no. 4, pp. 777–804, 1997.

[187] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “ Gradient-based learning applied to document recognition.,” *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998.

[188] T. Serre, L. Wolf, and T. Poggio, “Object recognition with features inspired by visual cortex,” *Computer Vision and Pattern Recognition*, vol. 2, pp. 994–1000, 2005.

[189] J. Mutch and D. Lowe, “Object class recognition and localisation using sparse features with limited receptive fields,” *International Journal of Computer Vision*, vol. 80, no. 1, pp. 45–57, 2008.

[190] T. Serre, M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman, and T. Poggio, “A Theory of Object Recognition: Computations and Circuits in the Feedforward Path of the Ventral Stream in Primate Visual Cortex,” Cambridge, USA, 2005.

[191] A. J. Yu, M. A. Giese, and T. A. Poggio, “Biophysiologically Plausible Implementations of the Maximum Operation,” *Neural Computation*, vol. 14, no. 12, pp. 2857–2881, 2002.

[192] L. Lampl, D. Ferster, T. Poggio, and M. Riesenhuber, “Intracellular Measurements of Spatial Integration and the MAX operation in complex cells of the cat primary visual cortex,” *Journal of Neurophysiology*, vol. 92, pp. 2704–2713, 2004.

[193] A. Lueschow, E. K. Miller, and R. Desimone, “Inferior Temporal Mechanisms for Invariant Object Recognition,” *Cerebral Cortex*, vol. 4, no. 5, pp. 523–531, 1994.

[194] T. Carlson, H. Hogendoorn, H. Fonteijn, and F. A. J. Verstraten, “Spatial coding and invariance in object-selective cortex,” *Cortex*, vol. 47, no. 1, pp. 14–22, 2009.

[195] R. M. Cichy, Y. Chen, and J. D. Haynes, “Encoding the identity and location of objects in human LOC.,” *Neuroimage*, vol. 54, no. 3, pp. 2297–2307, 2011.

[196] D. Lowe, “Object recognition from local scale-invariant features,” in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, vol. 2, pp. 1150–1157.

[197] D. Walther and C. Koch, “Modeling attention to salient proto-objects,” *Neural Networks, Elsevier*, vol. 19, no. 9, pp. 1395–1407, 2006.

[198] U. of Reading, “PETS 2009 Benchmark Data,” 2009. [Online]. Available: http://www.cvg.rdg.ac.uk/PETS2009/a.html.

[199] S. Lazebnik, C. Schmid, and J. Ponce, “Semi-Local Affine Parts for Object Recognition,” in *Proceedings of the British Machine Vision Conference*, 2004, pp. 959–968.

[200] S. Lazebnik, C. Schmid, and J. Ponce, “A Maximum Entropy Framework for Part-Biased Texture and Object Recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, vol. 1, pp. 832–838.

[201] S. Lazebnik, C. Schmid, and J. Ponce, “A Sparse Texture Representation Using Local Affine Regions.,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265–1278.

[202] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative models from few training examples: an incremental bayesian approach tested on 101 object cagories,” in *CVPR Workshop on Generative-Model Based Vision*, 2004.

[203] C. Parrage, R. Baldrich, and M. Vanrell, “Color Calibration.” [Online]. Available: http://www.cvc.uab.es/color_calibration/Database.html.

[204] C. A. Parraga, R. Baldrich, and M. Vanrell, “Accurate Mapping of Natural Scenes Radiance to Cone Activation Space: A New Image Dataset,” in *5th European Conference on Colour in Graphics, Imaging, and Vision - 12th International Symposium on Multispectral Colour Science*, pp. 50–57.

[205] T. Liu, J. Sun, J. Wang, N. Zheng, X. Tang, and H. Shum, “Learning to Detect A Salient Object,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 353–367, 2011.

[206] S. Arivazhagan and R. N. Shebiah, “Object Recognition using Wavelet Based Salient Points,” *The Open Signal Processing Journal*, vol. 2, pp. 14–20, 2009.

[207] L. Elazary and L. Itti, “A Bayesian model for efficient visual search and recognition,” *Vision Research*, vol. 50, no. 14, pp. 1338–1352, 2010.

[208] A. Borji and L. Itti, “Scene Classification with a Sparse Set of Salient Regions,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1902–1908.

[209] A. R. Webb, *Statistical pattern recognition*, 2nd Editio. Chichester, UK: John Wiley & Sons, 2002.

[210] I. M. Harris and P. E. Dux, “Orientation-invariant object recognition: evidence from repetition blindness,” *Cognition*, vol. 95, pp. 73–93, 2005.

[211] R. Guyonneau, H. Kirchner, and J. Thorpe, “Animals roll around the clock: The rotation invariance of ultrarapid visual processing,” *Journal of Vision*, vol. 6, pp. 1008–1017, 2006.

[212] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “SURF: Speeded Up Robust Features,” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[213] M. Hamidi and A. Borji, “Invariance analysis of modified C2 features: case study—handwritten digit recognition,” *Machine Vision and Applications*, vol. 21, no. 6, pp. 969–979, 2009.

[214] Y. Ke, “PCA-SIFT: a more distinctive representation for local image descriptors,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, pp. 506–513.

[215] K. Mikolajczyk, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630.

[216] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” *International Conference on Computer Vision & Pattern Recognition*, vol. 2, pp. 886–893.

[217] J. J. Yokono and T. Poggio, “Rotation Invariant Object Recognition from One Training Example,” 2004.

[218] A. Tsitiridis, P. Yuen, K. Hong, T. Chen, F. Kam, J. Jackman, D. James, and M. Richardson, “A biological cortex-like target recognition and tracking in cluttered background,” in *SPIE, Optics and Photonics for Counterterrorism and Crime Fighting*, 2009, vol. 7486, p. 74860G.

[219] A. Tsitiridis, P. Yuen, K. Hong, T. Chen, I. Ibrahim, J. Jackman, D. James, and M. Richardson, “An improved cortex-like neuromorphic system for target recognitions,” in *Remote Sensing SPIE Europe*, 2010.

[220] J. Zhang, T. Tan, and L. Ma, “Invariant Texture Segmentation Via Circular Gabor Filters,” in *16th International Conference on Pattern Recognition (ICPR’02)*, 2002, pp. 901–904.

[221] D. W. Nigel, “Goldfish Retina: Organization for Simultaneous Color Contrast,” *Science*, vol. 158, no. 3803, pp. 942–944, 1967.

[222] B. R. Conway, “Spatial structure of cone inputs to color cells in alert macaque primary visual cortex (V-1),” *Journal of Neuroscience*, vol. 21, no. 8, pp. 2768–2783, 2001.

[223] Y. Freund and R. Schapire, “A Short Introduction to Boosting,” *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 1, pp. 771–780, 1999.

[224] K. Barnard and P. Gabbur, “Color and Color Constancy in a Translation Model for Object Recognition,” in *Eleventh Color Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications*, 2003, pp. 364–369.

[225] G. Healey and D. Slater, “Global color constancy: recognition of objects by use of illumination-invariant properties of color distributions,” *Journal of the Optical Society of America A*, vol. 11, no. 11, pp. 3003–3010, 1994.

[226] B. V. Funt, K. Barnard, and L. Martin, “Is Machine Colour Constancy Good Enough?,” in *5th European Conference on Computer Vision*, 1998, pp. 445–449.

[227] X. Barnard, B. V. Funt, and L. Martin, “Colour constancy meets colour indexing,” School of Computer Science, Simon Fraser University, Canada, 2000.

[228] Y. Tsin, R. Collins, V. Ramesh, and T. Kanade, “Bayesian Color Constancy for Outdoor Object Recognition,” in *Computer Vision and Pattern Recognition (CVPR)* , 2001.

[229] S. Obdrzalek, J. Matas, and O. Chum, “On the Interaction between Object Recognition and Colour Constancy,” in *The International Conference on Computer Vision (ICCV03)*, 2003.

[230] C. Kanan, A. Flores, and G. W. Cottrell, “Color constancy algorithms for object and face recognition,” in *ISVC’10 Proceedings of the 6th international conference on Advances in visual computing - Volume Part I*, 2010.

[231] J. Van de Weijer, T. Gevers, and A. Gijsenij, “Edge-Based Color Constancy,” *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2207–2214, 2007.

[232] G. . H. Finlayson and E. Trezzi, “Shades of gray and colour constancy,” in *IS&T/SID Twelfth Color Imaging Conference*, 2004, pp. 37–41.

[233] K. E. A. Van de Sande, T. Gevers, and C. G. M. Snoek, “Evaluating Color Descriptors for Object and Scene Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[234] A. K. Jain and F. Farrokhnia, “Unsupervised Texture Segmentation Using Gabor Filters,” *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[235] P. Moreno, A. Bernardino, and J. Santos-Victor, “Gabor Parameter Selection for Local Feature Detection,” in *IBRIA - 2nd Iberian Conference on Pattern Recognition and Image Analysis*, 2005, pp. 11–19.

[236] J. K. Kamarainen, V. Kyrki, and H. Kalviainen, “Fundamental frequency gabor filters for object recognition.,” in *Proceedings of the 16th International Conference on Pattern Recognition.*, 2002, pp. 628–631.

[237] B. Gokberk, M. O. Irfanoglu, L. Akarun, and E. Alpaydın, “Selection of Location, Frequency and Orientation Parameters of 2D Gabor Wavelets for Face Recognition,” in *ASB’03 Proceedings of the 1st international conference on Advanced Studies in Biometrics*, 2003, pp. 138–146.

[238] B. Gokberk, M. O. Irfanoglu, L. Akarun, and E. Alpaydın, “Learning the best subset of local features for face recognition,” *Pattern Recognition*, vol. 40, pp. 1520–1532, 2007.

[239] P. Kruizinga, N. Petkov, and S. E. Grigorescu, “Comparison of texture features based on Gabor filters,” in *Proceedings of the 10th International Conference on Image Analysis and Processing*, 1999, pp. 142–147.

[240] T. P. Weldon, W. E. Higgins, and D. F. Dunny, “Efficient Gabor Filter Design For Texture Segmentation,” *Pattern Recognition*, vol. 29, no. 2, pp. 2005–2015, 1996.

[241] D. Zhang, A. Wong, M. Indrawan, and G. Lu, “Content-based Image Retrieval Using Gabor Texture Features,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 13–15, 2000.

[242] K. Laws, “Rapid texture identification,” in *Image Processing for Missile Guidance*, 1980, pp. 376–380.

APPENDICES

Appendix A HMAX tuning parameters

C1 layer			S1 Layer		
Scale band S	Spatial pooling grid (Ns x Ns)	Overlap Δs	Filter size s	Gabor σ	Gabor λ
1	8x8	4	7x7	2.8	3.5
			9x9	3.6	4.6
2	10x10	5	11x11	4.5	5.6
			13x13	5.4	6.8
3	12x12	6	15x15	6.3	7.9
			17x17	7.3	9.1
4	14x14	7	19x19	8.2	10.3
			21x21	9.2	11.5
5	16x16	8	23x23	10.2	12.7
			25x25	11.3	14.1
6	18x18	9	27x27	12.3	15.4
			29x29	13.4	16.8
7	20x20	10	31x31	14.6	18.2
			33x33	15.8	19.7
8	22x22	11	35x35	17.0	21.2
			37x37	18.2	22.8

Appendix B GBVS parameterisation

The following list of parameters is directly cited from the parameterisation MATLAB function within the GBVS software package including the original comments made from the creators [163]:

```
p.salmapmaxsize = 30;                % size of output saliency maps (maximum
dimension)
                                  % don't set this too high (e.g., >60)
                                  % if you want a saliency map at the
                                  % original image size, just used rescaled
                                  % saliency map
                                  % (out.master_map_resized in gbvs())

p.verbose = 1;                    % turn status messages on (1) / off (0)
p.verboseout = 'screen';          % = 'screen' to echo messages to screen
                                  % = 'myfile.txt' to echo messages to file

p.saveInputImage = 0;             % save input image in output struct
                                  % (can be convenient, but is wasteful
                                  %  to store uncompressed image data
                                  %  around)

%%%%%%%%%%%%% feature channel parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

p.channels = 'CIO';                  % feature channels to use encoded as a
string
                                  % these are available:
                                  %   C is for Color
                                  %   I is for Intensity
                                  %   O is for Orientation
                                  %   R is for contRast
                                  %   F is for Flicker
                                  %   M is for Motion
                                  % e.g., 'IR' would be only intensity and
                                  %       contrast, or
                                  %   'CIO'   would   be   only   color,int.,ori.
(standard)
                                  % 'CIOR' uses col,int,ori, and contrast

p.colorWeight = 1;                 % weights of feature channels (do not need
to sum to 1).
p.intensityWeight = 1;
p.orientationWeight = 1;
p.contrastWeight = 1;
p.flickerWeight = 1;
p.motionWeight = 1;
%p.gaborangles= [0 45 90 135];
p.gaborangles = [ 0 20 45 60 90 110 135 155 175 195 215 225 245 265 285 305];
% angles of gabor filters
p.contrastwidth = 1.9;               % fraction of image width = length of
square side over which luminance variance is
                                  % computed for 'contrast' feature map
                                  % LARGER values will give SMOOTHER
                                  %   contrast maps
```

```
p.flickerNewFrameWt = 1;          % (should be between 0.0 and 1.0)
                                  %   The   flicker   channel   is   the   abs()
difference
                                  % between the *previous frame estimate* and
                                  % current frame.
                                  % This parameter is the weight used
                                  % to update the previous frame estimate.
                                  % 1 == set previous frame to current
                                  %      frame
                                  % w == set previous frame to w * present
                                  %      + (1-w) * previous estimate
%p.motionAngles = [0 45 90 135];

p.motionAngles = [0 20 45 60 90 110 135 155 175 195 215 225 245 265 285 305];
                                  % directions of motion for motion channel
                                  %  --> 0 , /^ 45 , |^ 90 , ^\ 135 , etc.
                                  % question: should use more directions?
                                  % e.g., 180, 225, 270, 315, ?

%%%%%%%%%%%%%% GBVS parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

p.levels = [ 2 3 4];                % resolution of feature maps relative to
original image (in 2^-(n-1) fractions)
                                  % (default [ 2 3 4]) .. maximum level 9 is
allowed
                                  % these feature map levels will be used
                                  % if graph-based activation is used.
                                  % otherwise, the ittiCenter/Delta levels
                                  % are (see below)
                                  % minimum value allowed  = 2
                                  % maximum value allowed  = 9

p.multilevels = [];                  % [1 2] corresponds to 2 additional node
lattices ,
                                  % ... one at half and one at quarter size
                                  % use [] for single-resolution version of
algorithm.

p.sigma_frac_act = 0.15;          % sigma parameter in activation step of GBVS
(as a fraction of image width) - default .15
p.sigma_frac_norm = 0.06;           % sigma parameter in normalizaiton step of
GBVS (as a fraction of image width) - default .06
p.num_norm_iters = 1;             % number of normalization iterations in GBVS
- default 1

p.cyclic_type = 1;                   % use "2" for non-cyclic distance rules
(leads to "center bias" since nodes in center
                                  %  .. are more heavily connected.
                                  %  use  "1"  for  cyclic  distance  rules,
(eliminates center bias)

p.tol = .0001;                         % tol controls a stopping rule on the
computation of the equilibrium distribution (principal eigenvector)
                                  % the higher it is, the faster the algorithm
runs, but the more approximate it becomes.
```

```
                                % it is used by
algsrc/principalEigenvectorRaw.m - default .001


%%%%%%%%%% Parameters to use Itti/Koch and/or Simpler Saliency Algorith %%%%


p.useIttiKochInsteadOfGBVS = 0;     % use value '0' for Graph-Based Visual
Saliency
                                % use value '1' for Itti Koch algorithm:
                                % "A Model of Saliency-Based Visual
                                % Attention for Rapid Scene Analysis",
                                % PAMI 1998

p.activationType = 1;               % 1 = graph-based activation (default)
                                % 2 = center-surround activation (given
                                %     by ittiCenter/DeltaLevels below)
                                % ( type 2 used if useIttiKoch..= 1 )

p.normalizationType = 2;            % 1 = simplest & fastest. raises map values
to a power before adding them together (default)
                                % 2 = graph-based normalization scheme
                                % 3 = normalization by (M-m)^2, where M =
                                %     global maximum. m = avg. of local
                                %     maxima
                                % ( type 3 used if useIttiKoch..=1 )

p.normalizeTopChannelMaps = 0;      % this specifies whether to normalize the
                                % top-level feature map of each
                                % channel... (in addition to normalizing
                                % maps across scales within a channel)
                                % 0 = don't do it (default)
                                % 1 = do it. (used by ittiKoch scheme)


p.ittiCenterLevels = [ 2 3 4 ];    % the 'c' scales for 'center' maps
p.ittiDeltaLevels = [ 2 3 ];        % the 'delta' in s=c+delta levels for
'surround' scales
```

Appendix C Preparing MATLAB – only FHLib and Visual Salience

C.1 MATLAB FHLib (MFHLib)

From section 5.4 of this work and onwards, experiments on recognition are performed on an implementation of FHLib which was reprogrammed completely in MATLAB. The original program code from [**1**], contained a MATLAB interface but the majority of functions were created in MEX form using C/C++. This made the procedures run faster but in a less intuitive and clear way. Moreover, switching between programming languages, the lack of flow and comments made the code at places unreadable and overly specialised. The implementation in MATLAB here is not as fast as the original in computation time but it was created with simplicity in mind. It was also programmed in order to quickly facilitate the considerable amount of enhancements that were planned for the purposes of this thesis.

Following section 4.2.4, the programming steps are as followed:

1) Training Phase

a) Declare data directories for input datasets.

b) Declare global parameters, such as number of independent runs (in most experiments 3), total number of features, and total number of images per category. The total number of features is divided over the training images in order to identify the number of features per image for the extraction method.

c) Find training folders, path names and create library of features.

d) Load the first training image (numerical order) from the first training folder (alphabetical order) in unsigned integer 8 format initially and convert to double, scale image to 140 pixels at its shortest edge. Start extraction of features using the following parameters.

   i) Number of scales, default at 10.

   ii) Scale factor for the spatial pyramid, default $2^{(1/4)}$ or 1.2 approximately.

   iii) Gabor filter parameters (as in FHLib) set to default as 0.3, 5.6, 4.5 for γ, λ and σ respectively (sections 3.4.2, 4.2.4).

   iv) Gabor template size at 11x11 pixels.

v) Gabor orientation banks at 12 for FHLib.

e) Obtain the intensity image from the input training image using equation (4-38).

f) Create spatial pyramid for S1 layer.

g) Apply Gabor filters on the pyramid to produce S1 units.

h) Run a max-spatial pyramid of two scales (10x10 and 8x8) over the scales of the Gabor orientation pyramids in succession e.g. 1 - 2, 2 - 3, 3 - 4 etc, which result in 9 scales and in steps of 5 pixels (sub-sampling factor 2), to create S1 units.

i) Run inhibition over all scales using equation (4-44).

j) Merge orientation pyramids into one spatial pyramid (sparsification) and thus obtain the S2 layer units.

k) Start random extraction of patch sizes (4x4, 8x8, 12x12 and 16x16) from random co-ordinates across the image at randomly selected scales of the S2 layer. These patches along with their extraction co-ordinates and information (e.g. scale, size) constitute the templates or features stored in the library of step c) and their number depends from step b).

l) Loop over all training images and extract features following steps d)-k).

m) Having obtained the Feature Library all features are now used on the original training images (including the image from which the feature was extracted) as filters to extract their responses. More specifically, each feature is moved by 5 positions in all directions around the co-ordinates it was extracted from and by using the RBF equation (4-43) with sigma = 0.1, the maximum response is returned as a C2 vector. Thus, each S2 template yields a corresponding C2 vector.

i) After all C2 vectors for each image have been produced then having created

n) After all C2 vectors for the training images have been assembled then having created another variable with the class label for each, then the classifier, in this case a linear Support Vector Machine, is trained. The SVM is used with one-against-one strategy and with $C = \infty$, $\gamma = 1$, in all cases by default as the original FHLib. This setup has also been proven, using the cross-validation technique, to produce the best results [ref for toolbox].

o) As soon as the classifier is finished, the trained classifier along with the necessary variables and library of features is stored locally.

2) Testing Phase

a) Declare data directories for input test datasets. These paths are the same as the training datasets, since these folders contain all images. The testing images are instead initialised in the image immediately after the last training image. For example, if 30 images per category were chosen, then the first testing image is the image labelled as 31.

b) Steps d) – l) are identical but are instead employed for the testing images selected.

c) The stored features are now applied on the test image. Each feature is processed on the same coordinates as its original with 5 positions in all directions. Moreover, it is examined only on the same scale and not along the Gabor pyramid of the testing image at similar coordinates. The maximum responses of this procedure yield the C2 vectors for the testing image.

d) The test C2 responses are fed in the classifier which was trained during the training phase and classification accuracy results are produced.

C.2 Cranfield University Visual Salience

Saliency map extraction procedure for the orientation feature:

1. Input intensity image as in MFHLib, equation (6-4)
2. C centre pyramid scales at 1, 2, 3 and surround scales at 4, 5, 6, 7, 8, 9 and 10.
3. Scale input Gabor S1 maps generated from equation (6-5) (obtained from the shared process with MFHLib) for centre and surround scales.
4. Subtract maps according to equation (6-6).
5. Normalise maps.
6. Sum maps according to equation (6-7), and apply summation on the three resultant maps as obtained from each centre scale.
7. Run a Gaussian mask with a 5x5 window and $\sigma = 4$, over the saliency map to unify adjacent salient areas.

Saliency map extraction procedure for the colour feature:

1. Input RG and BY images obtained from equations (6-8) to (6-11).
2. Use centre pyramid scales at 1, 2, 3 and surround scales at 4, 5, 6, 7, 8, 9 and 10.
3. Scale input opponent maps generated for centre and surround scales.
4. Subtract maps according to equations (6-12) and (6-13).

5. Normalise maps.
6. Sum maps according to equations (6-14) and (6-15), and apply summation on the three resultant maps as obtained from each centre scale.
7. Apply saliency inhibition to remove “noise” information from the opponent maps and to promote the contrast of the colour distributions.
8. Run a Gaussian mask with a 5x5 window and $\sigma = 4$, over the saliency map to unify adjacent salient areas.